

# Diseño de un termostato Wi-Fi con cámara de fotos.



Grado en Ingeniería Eléctrica y Electrónica

## Trabajo Fin de Grado

Autor: Jefferson Quezada Álvarez

Director: Dr. Jesús Corres Sanz.

Pamplona, 30 de junio de 2015

## **RESUMEN**

El proyecto consiste en la elaboración de un prototipo de termostato que regule la temperatura de una vivienda y que se pueda controlar desde una aplicación de escritorio de Windows mediante Wifi. El termostato está formado por una tarjeta de desarrollo de Microchip, un módulo wifi para comunicarse con la aplicación del usuario y un módulo de cámara de fotos. La transferencia de información entre el termostato y la aplicación del usuario se realiza mediante un servidor a través de sockets.

Desde una pantalla LCD conectada al termostato se puede monitorizar la temperatura ambiente y la temperatura de referencia, que se puede variar desde pulsadores en el termostato. Además, desde la aplicación de escritorio se puede visualizar la temperatura ambiente, la de referencia, que también se puede modificar desde la aplicación, y se puede ver las fotos que envíe el termostato con la cámara de fotos.

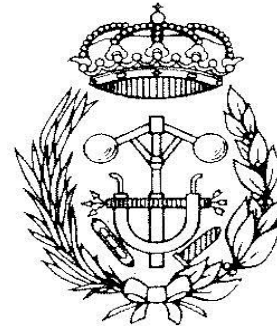
## **LISTA DE PALABRAS CLAVE**

- Termostato
- Termostato wifi
- Cámara de fotos
- Módulo wifi
- Wifi
- Sensor de temperatura
- Actuador
- Control de temperatura
- Monitorización
- Display LCD
- Explorer 16
- Domótica
- Aplicación de escritorio de Windows
- Programación C
- Programación C#
- Socket
- Arquitectura cliente - servidor

## ÍNDICE

MEMORIA .....	4
1. INTRODUCCIÓN .....	5
1.1 ANTECEDENTES .....	5
1.2 ESTADO DEL ARTE .....	6
1.3 MOTIVACIONES .....	8
1.4 OBJETIVO .....	9
2. RECURSOS EMPLEADOS .....	10
2.1 RECURSOS HARDWARE .....	10
2.2 RECURSOS SOFTWARE .....	14
3. CONTROL DE LA TEMPERATURA .....	16
3.1 SISTEMA .....	16
3.2 SENSOR DE TEMPERATURA.....	16
3.2.1 CARACTERÍSTICAS DEL SENSOR .....	16
3.2.2 CONVERSIÓN ANALÓGICA-DIGITAL. PROGRAMACIÓN .....	18
3.3 TEMPERATURA DE REFERENCIA.....	20
3.4 COMPARADOR Y CONTROLADOR SOFTWARE .....	21
3.5 ACTUADOR .....	23
4. CÁMARA DE FOTOS. MÓDULO C1098 .....	24
4.1 UART. COMUNICACIÓN SERIE .....	24
4.2 CONEXIÓN DEL MÓDULO C1098.....	25
4.3 PROGRAMACIÓN DEL MÓDULO C1098 .....	27
5. MÓDULO WIFI RN-171 .....	31
5.1 APLICACIÓN DE CONFIGURACIÓN DEL MÓDULO WIFI .....	33
6. COMUNICACIÓN TERMOSTATO-APLICACIÓN. ....	36
6.1 CONCEPTOS PREVIOS.....	36
6.1.1 MODELO TCP/IP .....	36
6.1.2 ARQUITECTURA CLIENTE-SERVIDOR .....	37
6.1.3 SOCKETS.....	37
6.1.4 CONCEPTOS DEL WI-FI.....	39
6.1.5 DIRECCION IP .....	40
6.2 ESTRUCTURA DE LA COMUNICACIÓN TERMOSTATO-SERVIDOR-USUARIO.	
SOLUCIÓN AL PROBLEMA DE IP DINÁMICA. ....	41

6.3	PROTOCOLO DE COMUNICACIÓN ENTRE TERMOSTATO-SERVIDOR-APLICACIÓN DE USUARIO.....	42
6.3.1	TRAMAS DE DATOS TERMOSTATO-SERVIDOR .....	43
6.3.2	TRAMAS DE DATOS APLICACIÓN DE USUARIO - SERVIDOR .....	44
7.	PROGRAMACIÓN DEL SERVIDOR.....	46
8.	PROGRAMACIÓN DE LA APLICACIÓN DE USUARIO .....	49
9.	PROGRAMACIÓN GENERAL DEL TERMOSTATO .....	51
10.	PROTOTIPO EN FUNCIONAMIENTO .....	54
11.	BIBLIOGRAFÍA.....	61



# MEMORIA

# 1. INTRODUCCIÓN

## 1.1 ANTECEDENTES

Históricamente, el ser humano ha tratado de controlar las circunstancias que le rodean y, entre ellas, el clima. Controlar las condiciones atmosféricas está fuera de nuestro alcance, pero sí que podemos controlar el clima dentro de nuestras viviendas.

Existen muestras de fuego prehistórico en el interior de las cuevas, quizá estos sean los primeros intentos de calefacción de la historia. Los griegos descubrieron el hipocausto y lo perfeccionaron los romanos, que los utilizaban en sus termas. Consistía en hacer fuego en una caldera apartada y conducir los humos por debajo del suelo para calentarlo, consiguiendo así el primer suelo radiante. A partir de la revolución industrial, se desarrolla la técnica de conducción de fluidos por tuberías, que han sido calentados en calderas. En el siglo XX, cuando Thomas Edison descubre la lámpara incandescente, se dio cuenta que la energía eléctrica se podía utilizar para producir calor.

Por otra parte, también se ha intentado regular la temperatura a una deseada. Alrededor de 1620, Cornelis Drebbel inventó un termostato de mercurio para regular la temperatura de una incubadora de pollos. Este es uno de los primeros dispositivos controlados por realimentación. El control del termostato moderno se desarrolló en la década de 1830 por Andrew Ure, que inventó el termostato bimetálico. El primer termostato ambiental eléctrico fue inventado por Warren S. Johnson en 1883. Albert Butz inventó el termostato eléctrico y lo patentó en 1886.

Los termostatos modernos usados en las viviendas son electrónicos. El elemento más habitual para medir la temperatura es un termistor, un dispositivo que cambia su impedancia en función de la temperatura. La regulación de la temperatura se hace con controladores PID (proporcional, integral y derivativo) y permiten programar una temperatura según la hora, el día de la semana, la eficiencia u otros eventos.

La domótica es un conjunto de sistemas capaces de automatizar una vivienda, por ello, un termostato es parte fundamental de la domótica, pues nos permite automatizar algo fundamental de una vivienda, la temperatura.

Más recientemente, se han empezado a comercializar termostatos programables remotamente. Los termostatos se conectan a la red wifi de la vivienda y se pueden controlar desde aplicaciones remotas vía internet, aplicaciones que pueden estar incluso en nuestro Smartphone.

## 1.2 ESTADO DEL ARTE

El uso de termostatos electrónicos para controlar la temperatura de las viviendas es muy popular en la actualidad, pero el uso de termostatos que se pueden controlar remotamente no está tan generalizado. Uno de los motivos puede ser el precio de estos dispositivos en el mercado actual, además de que se trata de un artilugio reciente en el mercado. Desde la popularidad de los Smartphone, son muchas las aplicaciones que pueden controlar distintos aspectos de la vivienda, como la temperatura.

En el mercado existen varios termostatos que se pueden controlar desde aplicaciones en nuestro Smartphone, en cualquier lugar del mundo, siempre que contemos con una conexión a internet. Uno que se ha hecho muy popular es el termostato Nest.

El termostato Nest se presentó el 25 de octubre de 2011 y tiene un precio en el mercado de alrededor de 250€. Es un producto que, mediante una combinación de sensores y algoritmos de aprendizaje es capaz de aprender el comportamiento y preferencias de los usuarios, funcionando de forma más autónoma cuanto más se utilice. Además, gracias a su conectividad wifi, el usuario es capaz de ajustarlo con su ordenador, smartphone o tablet y, mediante el uso de diferentes tecnologías, Nest es capaz de guiar al usuario a temperaturas con mayor eficiencia energética. También detecta cuando el usuario está en casa o no, para bajar la temperatura y seguir ahorrando energía.



*Figura 1. Termostato Nest. Fuente [www.nest.com](http://www.nest.com)*

Otra marca conocida es Honeywell, que comercializa varios termostatos wifi. Sus precios varían entre 150€ y 250€. Los más avanzados incluyen pantallas táctiles, control por voz para ajustar la temperatura y medición de la humedad ambiental.



Figura 2. Termostato Honeywell RTH9590WF. Fuente: [www.honeywell.com](http://www.honeywell.com)

Por último, cabe mencionar el termostato Netatmo. Es un termostato wifi controlado por una aplicación instalable en un smartphone. Además, controla la activación de la caldera con un relé por radiofrecuencia, sin la necesidad de que haya una conexión cableada entre el termostato y el relé. Su precio es de unos 170€.



Figura 3. Termostato Netatmo. Fuente: [www.netatmo.com](http://www.netatmo.com)



## 1.3 MOTIVACIONES

La domótica es el conjunto de sistemas capaces de automatizar una vivienda, aportando servicios de gestión energética, seguridad, confort y comunicaciones. Para que un sistema domótica se considere adecuado debe de disponer de soluciones para estas cuatro funciones básicas.

Un termostato wifi satisface tres de estas cuatro necesidades. Satisface la necesidad de confort, ya que existe un control automático de la temperatura, lo que produce sensación de comodidad y bienestar. También, satisface la comunicación, pues se puede gestionar la temperatura de la vivienda, tanto desde el interior con el termostato, como desde el exterior con la aplicación que permita conectarse vía internet al termostato wifi. Esta característica también se puede incluir en la necesidad de confort porque la gestión de la vivienda no queda limitada a su interior. Por último, también satisface la gestión energética, ya que existe una regulación automática de la temperatura, se puede programar para hacer un ajuste personalizado de costumbres de uso, permite monitorizar la temperatura de la vivienda de manera remota y permite ajustar la temperatura de referencia, lo que permite gestionar el consumo energético.

La única función que no realiza un termostato wifi sería la de seguridad. Las otras tres funciones, que están relacionadas unas con otras, sí que quedarían satisfechas. Si, además de gestionar la temperatura, el termostato puede capturar imágenes de la vivienda con una cámara de fotos que lleve incorporada, la necesidad de seguridad quedaría satisfecha. Desde la aplicación del usuario se podrían ver imágenes de la vivienda de manera remota, proporcionando seguridad.

Por lo tanto, un termostato wifi con una cámara de fotos incorporada satisfaría las cuatro funciones básicas que debe aportar una instalación domótica.

La motivación principal es el poder desarrollar una estación domótica que sea capaz de gestionar un aspecto fundamental de la vivienda, como es la temperatura, y que además, aporte soluciones para las cuatro funciones básicas de una instalación domótica. Otra gran motivación es el poder realizar un dispositivo cuyo desarrollo es bastante reciente en el mercado y cuya tecnología de comunicación también es muy actual.

Otra motivación es tratar de favorecer el ahorro energético, ya que con estos dispositivos se puede gestionar de una manera más inteligente la temperatura de una vivienda y, por tanto, el consumo energético.

## **1.4 OBJETIVO**

El objetivo de este trabajo final de grado es desarrollar un prototipo de termostato wifi que lleve incorporado una cámara de fotos y poder así satisfacer las cuatro funciones básicas de un sistema domótico: gestión energética, confort, comunicación y seguridad.

El control remoto vía wifi del termostato se realizará con una aplicación de escritorio de Windows. Desde esta aplicación se monitorizará la temperatura de la vivienda y se podrá asignar la temperatura de consigna. Además, se podrá ver la última foto que haya capturado la cámara de fotos.

Por último, desde el prototipo del termostato se podrá visualizar en una pantalla LCD la temperatura real y la de referencia, que se podrá modificar desde pulsadores en el termostato.

## 2. RECURSOS EMPLEADOS

El dispositivo que se va a desarrollar es un dispositivo electrónico, por tanto, son necesarios, tanto recursos hardware, como software.

### 2.1 RECURSOS HARDWARE

➤ EXPLORER 16:

La placa base utilizada para desarrollar el termostato es la Explorer 16, una tarjeta de desarrollo de Microchip. Además, es necesario el cable de alimentación de la Explorer 16, que se conecta directamente a la red. La Explorer 16 cuenta con las siguientes características:

- Adaptador para PIM (Processor Plug-In Modules), compatible con las versiones de PIM para los microprocesadores PIC24, PIC32 y dsPIC.
- Alimentación a  $V_{DC}$  de 9V, que proporciona +3,3V y +5V en la placa.
- Indicador LED de encendido.
- Puerto serie RS – 232
- Sensor térmico TC1047A integrado en la placa.
- Pantalla LCD de dos líneas y 16 caracteres.
- Cuatro pulsadores de propósito general y uno para resetear el microprocesador.
- Potenciómetro para una entrada analógica.
- Ocho indicadores LED.
- Multiplexores configurables para comunicaciones serie.
- Cristales independientes para la señal de reloj del microcontrolador (8MHz) y operación RTCC (32.768KHz).
- Conector para la tarjeta hija “PICKtail™ Plus card” de Microchip.
- Interfaz de 6 pines para el programador “PICKit™”
- Conectividad USB
- Conector RJ-11 para programador.
- Memoria serial EEPROM
- Área para prototipado.
- Conector JTAG.

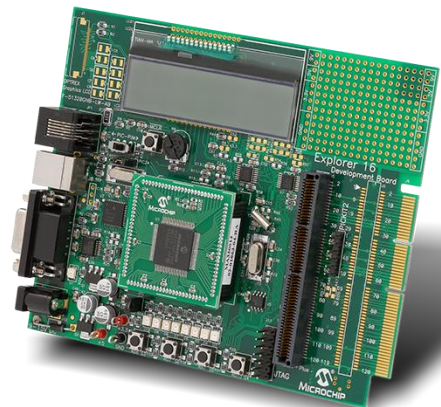


Figura 4. Tarjeta Explorer 16. Fuente: [www.microchip.com](http://www.microchip.com)

➤ PIC32MX360F512L PLUG-IN MODULE:

Es una pequeña tarjeta que puede ser usada en varias tarjetas de desarrollo. Es un sistema de conectar diferentes microprocesadores a una placa de desarrollo.

Esta tarjeta lleva un microcontrolador PIC32MX360F512L. Este micro presenta las siguientes características:

- 100 pines
- Frecuencia de hasta 80 MHz
- Memoria de programa de 512 KB
- Memoria de datos de 32 KB
- 5 Timers / 5 capture / 5 compare
- 16 canales de ADC (Analogic - Digital Converter) de 10 bit
- 4 canales programables de DMA (Direct Memory Access)
- VREG
- Trace
- 2 UART / 2 SPI / 2 I<sup>2</sup>C
- 2 Comparadores
- PMP / PSP
- JTAG



Figura 5. Módulo plug-in del PIC32MX360F512L.

Fuente: <http://www.liveshop.lv/>

➤ TARJETA HIJA PARA PROTOTIPO PICTail PLUS:

Es una tarjeta de prototipado de 8 x 8 cm que se conecta al conector PICTail Plus de la Explorer 16. Proporciona acceso a todos los pines del microprocesador.



Figura 6. Tarjeta hija de prototipado PICTail Plus. Fuente: <http://store.comet.bg/>

➤ SENSOR DE TEMPERATURA TC1047A

Es un sensor de temperatura que es capaz de medir temperaturas entre  $-45^{\circ}\text{C}$  y  $+125^{\circ}\text{C}$ . Este sensor ya está integrado en la tarjeta Explorer 16 y se encuentra conectado al pin AN4 del microcontrolador.

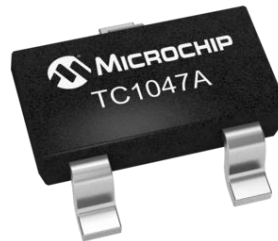


Figura 7. Sensor de temperatura T1047A. Fuente: [www.microchip.com](http://www.microchip.com)

La alimentación del sensor puede ser de entre 2,7V y 5,5V. La salida es analógica y directamente proporcional a la temperatura medida.

➤ MODULO WIFI RN-171-EK

Es una tarjeta de evaluación para el módulo RN-171. Se puede utilizar para agregar conectividad Wi-Fi a aplicaciones embebidas. La comunicación con el sistema al que se conecta es a través de UART (Universal Asynchronous Receiver-Transmitter) a niveles de tensión TTL (Transistor-Transistor Logic). Sus características son las siguientes:

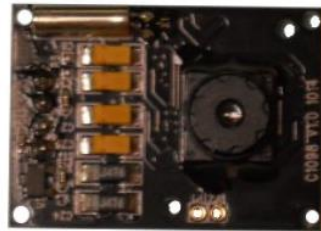
- 2,4GHz IEEE 802.11b/g
- Aplicaciones de red como TCP/IP, DHCP, DNS, ARP, ICMP, UDP, Telnet, cliente FTP y cliente HTML.
- Configurable por Wi-Fi o por UART usando comandos ASCII
- Alimentación a 3,3V por 2 pilas AAA, o a 5V a través de cable USB
- 10 pines I/O de propósito general
- 8 interfaces de sensor analógico.
- Antena incluida en la tarjeta de evaluación.
- Pulsadores para reset y para aplicaciones de lanzamiento.



Figura 8. Tarjeta de evaluación del módulo wifi RN-171. Fuente: [www.microchip.com](http://www.microchip.com)

➤ MODULO DE CÁMARA DE FOTOS C1098

Es un módulo de cámara de fotos que lleva incorporado un compresor JPEG. Se alimenta entre 3V y 3,6V. La comunicación con el sistema al que se conecta es a través de UART a niveles TTL. Los comandos de configuración se hacen a 14400bps, pero el resto de comunicación se puede hacer hasta 460,8Kbps a través de la UART. Las fotos que captura pueden tener una resolución QVGA (Quarter Video Graphics Array) o VGA (Video Graphics Array).



*Figura 9. Módulo de cámara de fotos C1098. Fuente: [www.electronics123.net/](http://www.electronics123.net/)*

➤ DEBUGGER REAL ICE

El depurador utilizado para programar la tarjeta Explorer 16 es el REAL ICE de Microchip. Con este depurador se pueden identificar y corregir errores de programación. También permite ejecutar el programa paso a paso y colocar “breakpoints”, es decir, líneas del programa donde se detendrá su ejecución, permitiendo examinar el estado actual de las variables.

Además del REAL ICE, también son necesarios un cable USB para conectarlo al ordenador y un cable RJ-11 para conectarlo a la Explorer 16.



*Figura 10. Debugger REAL ICE. Fuente: [www.microchip.com](http://www.microchip.com)*

➤ OTROS RECURSOS HARDWARE:

Además de los nombrados anteriormente, se van a utilizar en este proyecto los siguientes elementos:

- Un ordenador común de sobremesa, que cuenta con conectores USB y del puerto serie, necesarios para el desarrollo del proyecto.
- Integrado LP2951
- Resistencias y condensadores de diferentes valores.
- Cables para conectar los diferentes módulos.

## 2.2 RECURSOS SOFTWARE

➤ MPLABX IDE:

Es un programa de software que se ejecuta con los sistemas operativos Windows, Mac OS, o Linux para desarrollar aplicaciones para microcontroladores Microchip y controladores de señales digitales. Se llama entorno de desarrollo integrado (IDE) porque proporciona un único medio integrado para desarrollar microcontroladores embebidos.

➤ COMPILADOR MPLAB XC32:

Es un compilador optimizado que traduce los programas ANSI C en lenguaje fuente de dispositivos de 32 bits.

➤ VISUAL STUDIO 2013:

Es una plataforma de desarrollo integrado que permite crear aplicaciones para Windows, Android e iOS, además de aplicaciones web y servicios de nube. Permite programar con los siguientes lenguajes: C#, Visual Basic, F#, C++, Python, Node.js y HTML/JavaScript.

➤ DESIGNSPARK PCB 7.0:

Es un software libre que permite hacer esquemáticos y diseño electrónico. Está destinado tanto para diseñadores profesionales, como para aficionados, educadores y estudiantes.

➤ DOCKLIGHT SCKIPTING V2.0:

Es una herramienta de test y de simulación que permite interactuar tanto con el puerto serie del ordenador en el que se encuentre instalado, como con la red LAN (Local Area Network) a la que se encuentre conectado, es decir, abrir una comunicación haciendo de servidor o de cliente con una dirección IP y un puerto específico.

➤ TERA-TERM:

Es una herramienta similar a Docklight, con la que también se puede crear archivos con los datos que recibe, ya sean binarios o ASCII.



### 3. CONTROL DE LA TEMPERATURA

La estrategia de control de la temperatura se realiza con el sistema en lazo cerrado (realimentado) que se refleja en el siguiente diagrama de flujo. La salida del sistema (temperatura real) es medida por el sensor, la temperatura medida por el sensor es comparada con la temperatura de referencia, generando un error. Este error es la entrada al controlador, que se encarga de generar una acción de control en función del error de entrada. La acción de control es adaptada por el actuador, que genera la acción que el sistema es capaz de procesar.

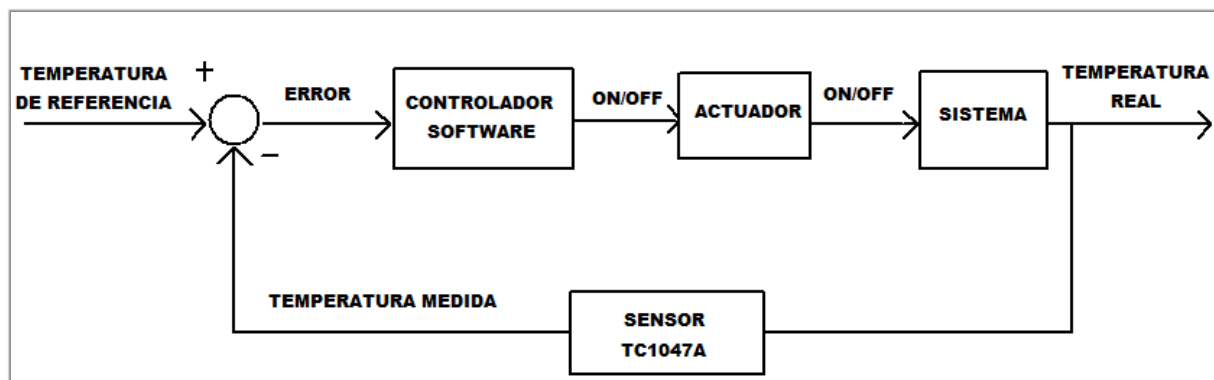


Figura 11. Diagrama de flujo del control de la temperatura.

#### 3.1 SISTEMA

El sistema a controlar sería el calentador de una vivienda que se pueda controlar con la orden de encendido o apagado. En el caso de este proyecto, no se utiliza ningún sistema calentador. Para poder ver de forma visual la orden que está enviando el termostato, se indica con un LED de la Explorer 16. Además, en la pantalla LCD se muestra "OFF" si el termostato está enviando la orden de apagado y "ON" si está enviando la orden de encendido.

#### 3.2 SENSOR DE TEMPERATURA

##### 3.2.1 CARACTERÍSTICAS DEL SENSOR

El sensado de la temperatura se realiza con el sensor de temperatura que lleva integrado la placa de desarrollo Explorer 16. Se trata del sensor TC1047A, un integrado que convierte la temperatura medida en una tensión de salida analógica. El rango de medición es de entre  $-40^{\circ}\text{C}$ , cuya tensión de salida es 100mV, y  $+125^{\circ}\text{C}$ , cuya tensión de salida es de 1750mV típicamente.

El incremento de la tensión de salida frente a la temperatura es de 10mV/°C tal y como se puede observar en la gráfica de abajo proporcionada por el fabricante. La tensión de salida típica a 0°C es 500mV por lo que la tensión de salida en función de la temperatura seguirá la siguiente ecuación:

$$V_{OUT} = (10\text{mV}/^{\circ}\text{C}) \cdot (\text{Temperatura } ^{\circ}\text{C}) + 500\text{mV} \quad [1]$$

A partir de esta ecuación se puede obtener el valor de la temperatura medida en función de la tensión de salida del sensor con la siguiente ecuación:

$$\text{Temperatura } (^{\circ}\text{C}) = \frac{V_{OUT} - 500\text{mV}}{10\text{mV}/^{\circ}\text{C}} \quad [2]$$

Así pues, no es necesaria ninguna etapa de acondicionamiento de la señal de salida del sensor porque la tensión que genera es lineal en función de la temperatura y tiene un rango de tensión de salida que es suficiente para ser convertida por el microcontrolador.

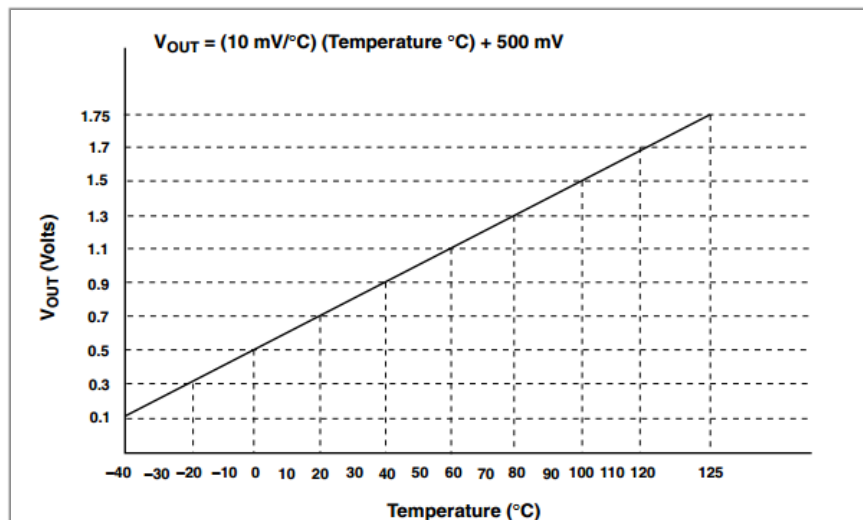


Figura 12. Tensión de salida del sensor T1047A frente a la temperatura. Fuente: Datasheet del sensor.

En la siguiente imagen se puede observar el esquema del montaje del sensor de temperatura en la Explorer 16.

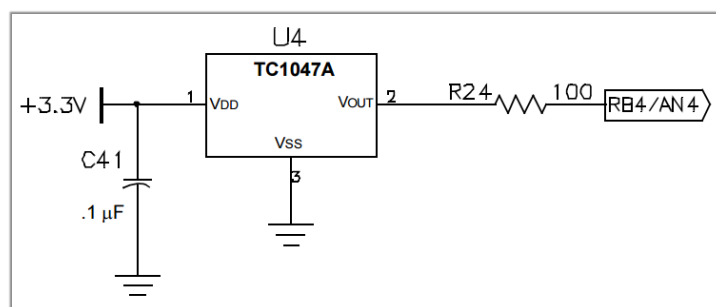


Figura 13. Esquema del montaje del sensor de temperatura en la Explorer 16. Fuente: Guía de usuario de la Explorer 16.

### 3.2.2 CONVERSIÓN ANALÓGICA-DIGITAL. PROGRAMACIÓN

Dado que la señal que genera el sensor es analógica, es necesario que el microcontrolador convierta esta señal analógica en valores digitales. Para ello, se utilizará el conversor analógico-digital (ADC) del micro.

Para convertir una señal analógica en digital es necesario muestrear la señal. Al mostrar la señal se produce una pérdida de la información porque el periodo de tiempo en el que no se toma una muestra, no se está obteniendo información de la señal analógica. Si el periodo de muestreo es demasiado grande, la señal digital obtenida no se correspondería a la señal analógica de entrada, no se podría reconstruir esta señal debido a la pérdida de información. A este efecto se le conoce como aliasing.

Según el teorema del muestreo de Nyquist-Shannon, la reconstrucción de una señal periódica continua en banda base es matemáticamente posible si la señal está limitada en banda y la frecuencia de muestreo es superior al doble de su ancho de banda.

En el caso del sensado de la temperatura, la frecuencia de variación de la señal de salida del sensor será muy pequeña, ya que la temperatura no puede variar bruscamente a altas frecuencias. Por este motivo, el efecto del aliasing no se tiene en cuenta porque el periodo de muestreo elegido es de un segundo, periodo en el cual, el valor de la temperatura apenas puede variar.

Además de la pérdida de la información, una señal digital está cuantizada, es decir, no puede representar todos los valores, como lo haría una señal analógica. La resolución del ADC del microcontrolador es de 10 bits. Esto quiere decir que es capaz de reproducir la señal analógica con un valor entero entre 0 y 1024 ( $2^{10}$ ).

La tensión de referencia utilizada por el conversor como fondo de escala (FS) es 3,3V. Así pues, el valor que se obtendrá del conversor será 1024 cuando la tensión analógica de entrada sea 3,3V. La resolución dada en mV será:

$$\text{Resolución del ADC (mV)} = \frac{FS}{2^N} = \frac{3,3}{1024} = 3.223 \text{ mV}$$

El valor que se obtiene de la conversión será el múltiplo de 3.223mV más cercano al valor muestreado. En la siguiente imagen se puede observar una señal analógica en azul, una cuantizada en verde, una muestreada con puntos y una señal digital con asteriscos.

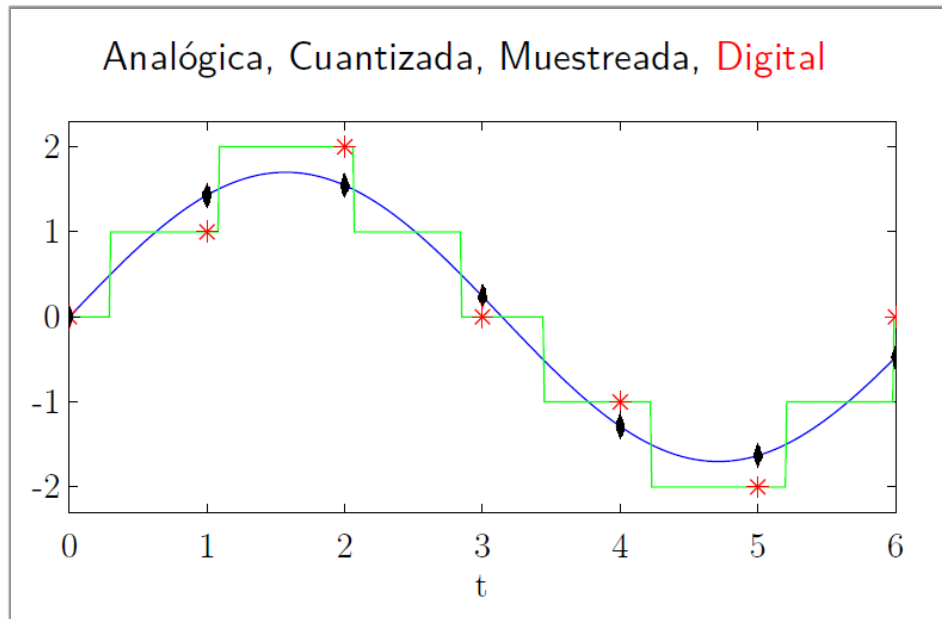


Figura 14. Representación de una señal analógica, una cuantizada, una muestreada y una digital.

Dado que la resolución del ADC del microcontrolador en mV es de 3,223mV y la variación de la tensión de salida del sensor de temperatura varía 10mV/°C, quiere decir que la resolución de este conversor en °C será:

$$\text{Resolución del ADC (}^{\circ}\text{C)} = \frac{1^{\circ}\text{C}}{10\text{mV}} \cdot 3,223\text{mV} = 0,3223^{\circ}\text{C}$$

La resolución del ADC del microcontrolador es de 3,223mV, que corresponden a 0,3223°C, un valor bastante pequeño tratándose de temperatura, por lo que el conversor del PIC32 se considera suficiente para el sensado de la temperatura.

La parte del programa encargada del sensado de la temperatura consiste en realizar una conversión de la señal de entrada procedente del sensor de temperatura, con una frecuencia de muestreo de 1 segundo.

El valor de la temperatura en °C se calcula en el programa a partir del valor obtenido de la conversión de la siguiente forma:

$$V_{ADC} \text{ (mV)} = \frac{3300\text{mV}}{2^{10}} \cdot \text{valor conversión} \quad [3]$$

$$\text{Temperatura (}^{\circ}\text{C)} = \frac{V_{ADC} - 500\text{mV}}{10\text{mV}/^{\circ}\text{C}} \quad [4]$$

Este valor de temperatura es el que se compara con el valor de temperatura de referencia en el diagrama de flujo utilizado como estrategia de control.

### 3.3 TEMPERATURA DE REFERENCIA

La temperatura de referencia es la temperatura a la que se desea que se encuentre la salida del sistema (temperatura de la vivienda). La temperatura de referencia es un variable entera del programa del termostato. Puede ser modificada, tanto por pulsadores en el termostato, como desde la aplicación del cliente de manera remota gracias al módulo wifi. Este funcionamiento se verá más adelante. En este apartado sólo se explicará la modificación de la variable de temperatura de referencia desde el termostato.

Para variar la temperatura desde el termostato se han utilizado los pulsadores de la tarjeta Explorer 16. El pulsador S3 se utiliza para aumentar en uno el valor de la temperatura de referencia, y el pulsador S6 se utiliza para disminuir en uno su valor.

Desde la pantalla LCD se puede visualizar la temperatura real de forma predeterminada y también se puede visualizar la temperatura de referencia cuando se presiona uno de los pulsadores que la modifican. Además, el pulsador S5 se utiliza para ver la temperatura de referencia sin modificarla, simplemente para visualizarla.

En la siguiente imagen se puede observar el esquema del montaje de los pulsadores en la tarjeta Explorer 16.

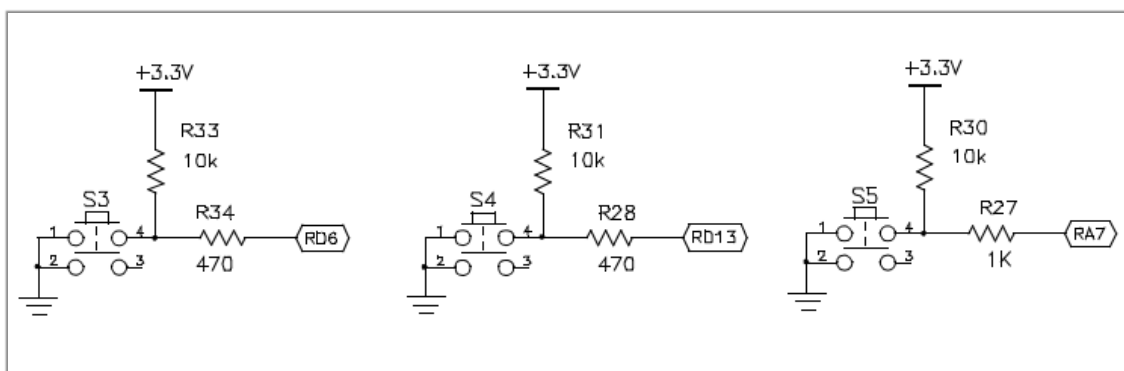


Figura 15. Esquema del montaje de los pulsadores utilizados en la Explorer 16. Fuente: Guía de referencia de la Explorer 16.

Según el esquemático, cuando el pulsador está presionado, la entrada digital a la que está conectado el pulsador se pone a cero. En reposo la entrada está a nivel alto.

A la hora de programar los pulsadores, se ha tenido en cuenta el efecto de los rebotes cada vez que se pulsan o se deja de pulsar. La siguiente gráfica muestra el efecto de estos rebotes en la señal de entrada al microcontrolador.

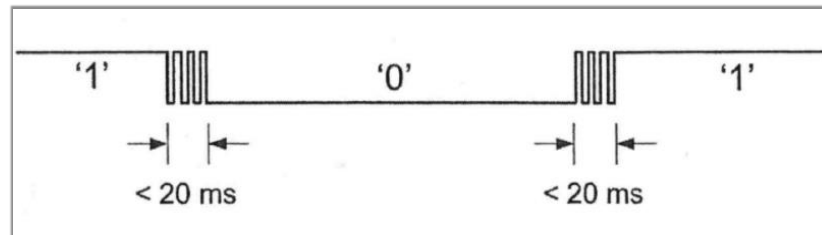


Figura 16. Efectos de los rebotes en los pulsadores.

Para evitar la problemática de los rebotes se ha optado por una solución software. En el programa, cada vez que se detecta que algún pulsador ha sido pulsado, se incluye un retardo mínimo de 20ms y se vuelve a comprobar si sigue pulsado, si es así, se realiza la parte del programa asociada al pulsador, sino, no la realiza.

### 3.4 COMPARADOR Y CONTROLADOR SOFTWARE

Tanto el controlador del sistema, como el comparador son elementos software y están relacionados. Ambos forman parte del programa del termostato.

Dado que la entrada del sistema a controlar es de tipo on/off, la salida del controlador también será on/off. Como el control se realiza por software en el microcontrolador, la salida será un pin en estado alto o bajo.

Se ha optado por un control por histéresis para evitar el encendido y apagado continuo que provocan las perturbaciones cuando la temperatura medida cruza el valor de la temperatura de referencia. Con este tipo de control, la salida del sistema a controlar será siempre oscilante en régimen permanente, pero acotada. Sin embargo, tratándose de control de temperatura, se considera un control suficiente.

En el control por histéresis se establecen dos límites en los que la temperatura va a oscilar, uno por encima de la temperatura de referencia y otro por debajo. Normalmente ambos límites se encuentran a la misma distancia de la referencia. Este tipo de control se basa en crear un doble comportamiento del sistema dentro del rango en el que oscilará la temperatura. Como se puede ver en la siguiente imagen, obedecerá la orden de encendido cuando se esté incrementando la temperatura, y la orden de apagado cuando, al alcanzar el límite superior, disminuya la temperatura, hasta que llegue al límite inferior, donde obedecerá la orden de encendido y empezará a aumentar de nuevo la temperatura.

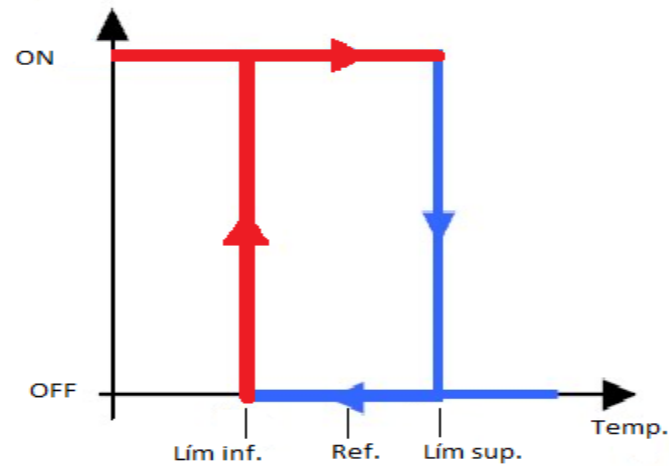


Figura 17. Representación del control por histéresis.

En la siguiente imagen se muestra la evolución de la temperatura medida frente a tiempo, con un control por histéresis para una temperatura de referencia dada. En la figura se puede observar la oscilación acotada entre los límites de histéresis de la temperatura en régimen permanente.

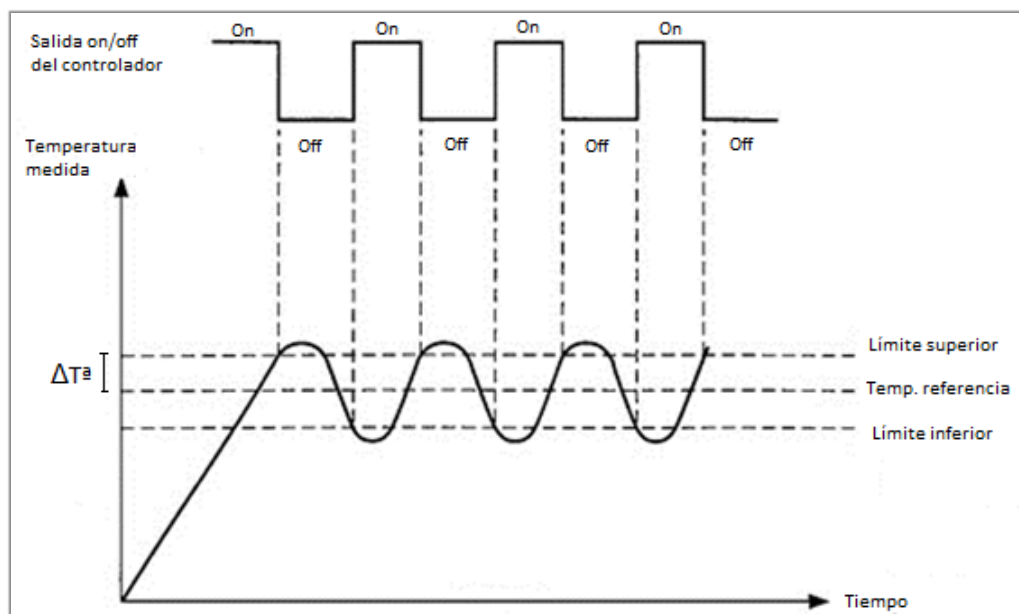


Figura 18. Evolución de la temperatura medida con un control por histéresis.

Para realizar el control por histéresis en el programa del termostato, el controlador cambia la orden a encendido sólo cuando la temperatura de referencia es mayor que la temperatura medida mas el límite del incremento de temperatura ( $\Delta T^a$ ). En cambio, la orden solo cambia a apagado cuando la temperatura de referencia es menor que la temperatura medida menos el límite del incremento de temperatura ( $\Delta T^a$ ).

### 3.5 ACTUADOR

Dado que no se actúa sobre un sistema calentador, no se ha construido un circuito como actuador, pero se propone un circuito que puede cubrir esta función.

El calentador sobre el que se actuaría se alimenta a la tensión de red, sin embargo, la orden de encendido y apagado son señales digitales de salida del microcontrolador (señales de 3,3V y 0V) por lo que es necesario un circuito que reciba la orden del controlador y actúe consecuentemente con el calentador.

Es necesario aislar eléctricamente el termostato del calentador, porque éste funciona con potencias mucho más grandes que el termostato y cualquier flujo de corriente procedente del calentador lo estropearía. Para aislar los circuitos se utiliza un optotriac. Este integrado lleva un diodo led que se activa con la señal de activación del microcontrolador. Cuando el led emita luz, se estimulará la puerta del triac que lleva el optotriac y comenzará a conducir. La corriente por el led tiene que ser lo suficiente como para garantizar que se estimulará la puerta del triac. La salida del triac será la que se conecte a la parte de potencia. Con este modo de funcionamiento la parte de potencia se comunica con la parte de señal a través de una luz, de modo que no habrá contacto eléctrico entre ambas partes y, por consiguiente, tampoco habrá fugas de corriente. Además, el integrado lleva un detector de paso por cero para determinar el momento en que la tensión alterna se hace cero. De esta manera se alimentará a la carga con una tensión inicial cero que irá aumentando de forma gradual. Este procedimiento se hace necesario especialmente con cargas inductivas.

Para el encendido y apagado del calentador se utiliza otro triac, pero éste será uno capaz de disipar la potencia que circule por él. Este triac se activará con la salida del optotriac. En la siguiente imagen se puede ver el circuito propuesto como actuador.

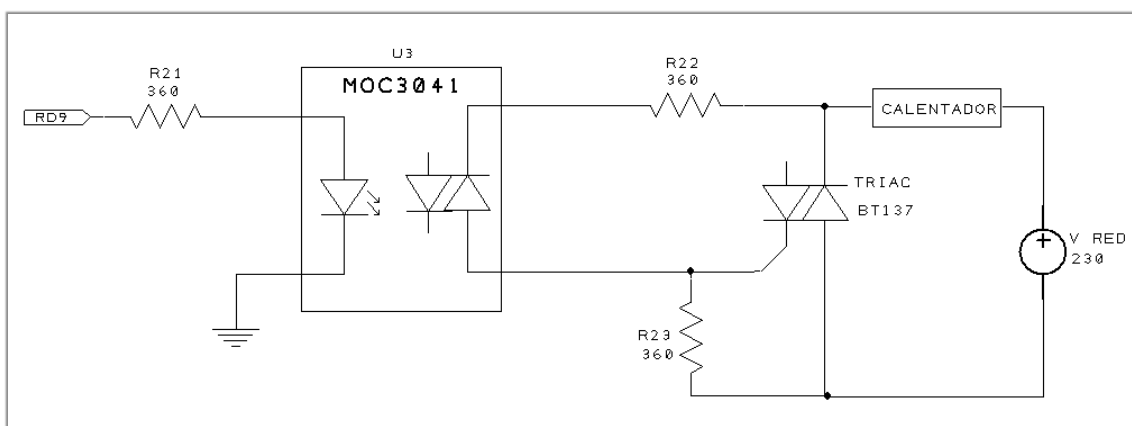


Figura 19. Esquema del circuito propuesto como actuador.



## 4. CÁMARA DE FOTOS. MÓDULO C1098

El módulo C1098 es un sensor de imagen que lleva integrado un compresor JPEG (Joint Photographic Experts Group). También, lleva integrado una interfaz UART (Universal Asynchronous Receiver-Transmitter) para la comunicación con el dispositivo que lo gobierne.

### 4.1 UART. COMUNICACIÓN SERIE

El transmisor-receptor asíncrono universal (UART) es un dispositivo que controla los puertos y dispositivos serie. Su función principal es convertir los datos serie en paralelos, cuando se trata de datos recibidos, y de convertir datos paralelos a serie cuando se trata de datos a transmitir. En definitiva, genera la comunicación serie.

La comunicación serie consiste en el envío de un bit de datos de manera secuencial, es decir, un bit a la vez y a una velocidad de transmisión igual entre el emisor y el receptor. En cambio, en la comunicación paralela, se envían múltiples bits de manera secuencial, pero se requiere una gran cantidad de hilos conductores, un hilo por cada bit.

En la comunicación asíncrona, el proceso de sincronización entre el emisor y el receptor se realiza en cada palabra de código transmitido. No existe una línea de reloj común que establezca la duración de un bit y así se pueda enviar el dato en cualquier momento. Por ello, previo al inicio de la comunicación, el receptor y el emisor deben establecer una misma velocidad de transmisión. Cada byte a transmitir debe ir con un indicador de inicio y de fin de byte. El inicio de un byte de datos a transmitir se indica con un bit de inicio y el final con uno o dos bits de parada. Además, antes del indicador de parada, se puede incluir un bit de paridad, que indica si el número de bits transmitidos es par o impar, para verificar errores. En la siguiente figura se pueden ver las distintas partes de una trama en una comunicación serie asíncrona.

La configuración mas utilizada es un bit de parada y no usa bit de paridad. Las velocidades de transmisión típicas son: 9600bps, 14400bps, 19200bps, 38400bps, 57600bps, 115200bps, etc. Se utiliza un hilo para la transmisión de datos y otros para la recepción llamados normalmente TX y RX respectivamente.

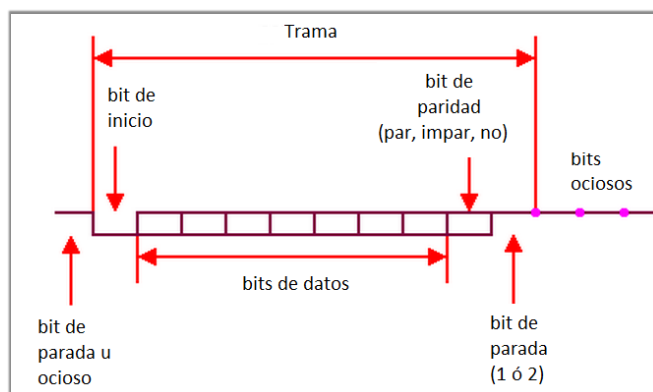


Figura 20. Trama de la comunicación serie asíncrona.

Los niveles de tensión utilizados pueden ser niveles RS-232, donde el nivel lógico 1 es entre -3V y -15V, y el nivel lógico 0 está entre +3V y +15V. También, pueden ser niveles de tensión TTL (Transistor-Transistor Logic), donde el nivel lógico 1 es comúnmente 5V ó 3,3V, y el nivel lógico 0 es 0V.

## 4.2 CONEXIÓN DEL MÓDULO C1098

El microcontrolador utilizado tiene dos UARTs, ambas a niveles TTL de 3,3V. La UART del módulo de cámara de fotos también es de niveles TTL a 3,3V, por lo que se pueden conectar directamente sin la necesidad de un hardware intermedio.

La documentación del módulo indica que solo son necesarios cuatro hilos para su funcionamiento. Dos para la alimentación, uno a 3,3V y otro para masa, y los otros dos para la transmisión y recepción de datos, TX y RX.

La interfaz UART del microcontrolador utilizada para la comunicación con el módulo C1098 es la UART 1. Hay que tener en cuenta que el pin de transmisión de la UART del micro (U1TX) se tiene que conectar con la recepción del módulo (RX) y el pin de transmisión del módulo (TX) se tiene que conectar pin de recepción de la UART1 del micro (U1RX).

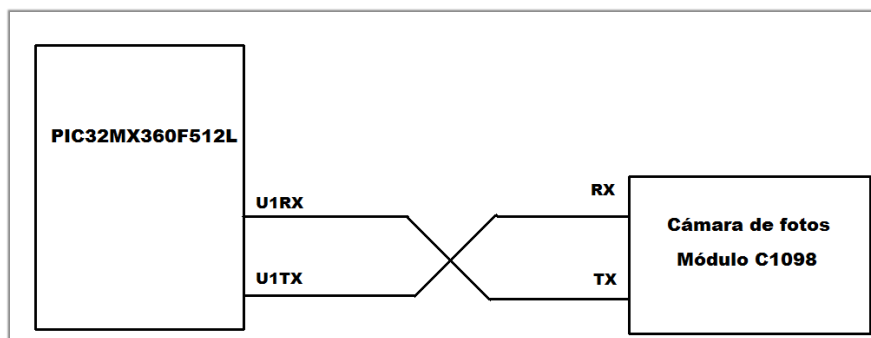


Figura 21. Esquema de la conexión del módulo C1098 con el PIC32MX360F512L.

Para alimentar el módulo de cámara de fotos no se ha utilizado directamente una fuente de 3,3V porque se va a controlar el encendido y apagado de la cámara desde el micro. De esta forma se consigue reducir el consumo de energía, apagando la cámara cuando no se use.

Para poder controlar la activación de la cámara se ha utilizado el integrado LP2951 que proporciona 3,3V controlados con una señal digital.

El circuito montado para utilizar el integrado el que se recomienda para su uso en la hoja de características del integrado.

El modo de funcionamiento del circuito es el siguiente: cuando el pin de control (RC4) está a nivel bajo, el integrado da como salida 3,3V, por lo que la cámara estará encendida; cuando RC4 esté a nivel alto, el integrado da como salida 0V y la cámara estará apagada.

En la siguiente imagen se puede ver el esquema del circuito utilizado para controlar la alimentación del módulo de cámara de fotos.

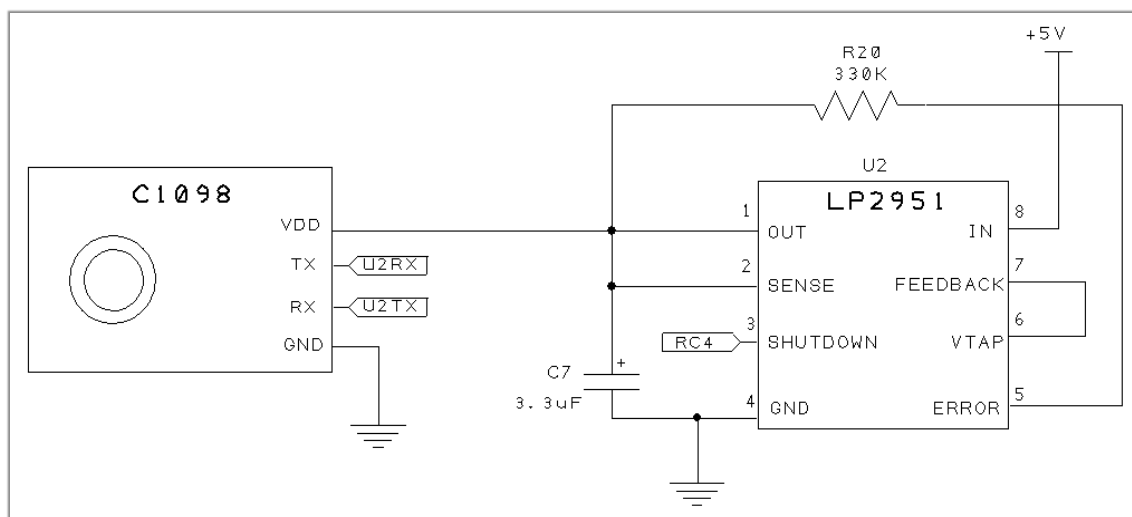


Figura 22. Esquema del diseño de control de la alimentación del módulo C1098.

### 4.3 PROGRAMACIÓN DEL MÓDULO C1098

En la hoja de características del módulo se explica el protocolo de comunicación que hay que seguir para obtener una foto de la cámara. Se podría dividir en tres partes: la sincronización con la cámara, la configuración y la captura de foto.

En la etapa de sincronización se envían comandos de sincronización (SYNC) hasta que la cámara responde correctamente con el comando de asentimiento (ACK) correcto y con un comando de sincronización de vuelta. En esta etapa, la velocidad de transmisión tiene que ser de 14400bps, ya que ésta es la velocidad por defecto de la cámara. Si no responde correctamente en 60 intentos hay que resetear el módulo.

Una vez que está sincronizada, se pasa a la etapa de configuración. En esta etapa se envían dos comandos de configuración. El primero es el "Initial" y sirve para establecer la velocidad de transmisión que se desee y para ajustar la resolución de la fotografía a QVGA o VGA. El segundo es el "Set Package Size" y sirve para establecer el tamaño de los paquetes en los que la cámara enviará los datos de la imagen. Para ambos comandos, la cámara debe de responder correctamente con el comando ACK que corresponda, sino, hay que enviar un comando de Reset y, si tampoco responde correctamente a éste, hay que cortar la alimentación e iniciar de nuevo todo el proceso.

Una vez configurado el módulo, se pasa a la etapa de captura de imagen. Se envían dos comandos, uno llamado "Snapshot" y el otro "Get Picture". En ambos casos, tiene que responder con el ACK que corresponda, sino, hay que proceder como anteriormente se ha explicado. Después, la cámara envía un comando llamado "Data Length" que contiene el número de bytes del que está compuesta la fotografía. Al responder al módulo con un ACK, éste envía un paquete de datos. Cuando se ha recibido todo el paquete de datos, se envía un ACK para recibir el siguiente paquete de datos y así sucesivamente. En cada comando de ACK enviado, hay que aumentar el valor de dos de sus bytes, que sirven como contador de paquetes. Con el último paquete de datos, estos bytes contador se tienen que enviar con el valor F0F0h para indicar que se ha recibido el último paquete de datos, tal y como indica la hoja de características.

Hay que tener en cuenta que al cambiar la velocidad de transmisión de la cámara con el comando "Initial", también hay que cambiar la de la UART1 de microcontrolador, sino, no podrán interpretar correctamente los datos que se envían.

Para programar el módulo, se ha seguido este protocolo de comunicación, que es el que corresponde con el diagrama de flujo que se muestra en la siguiente imagen, pero se han realizada ciertos cambios.

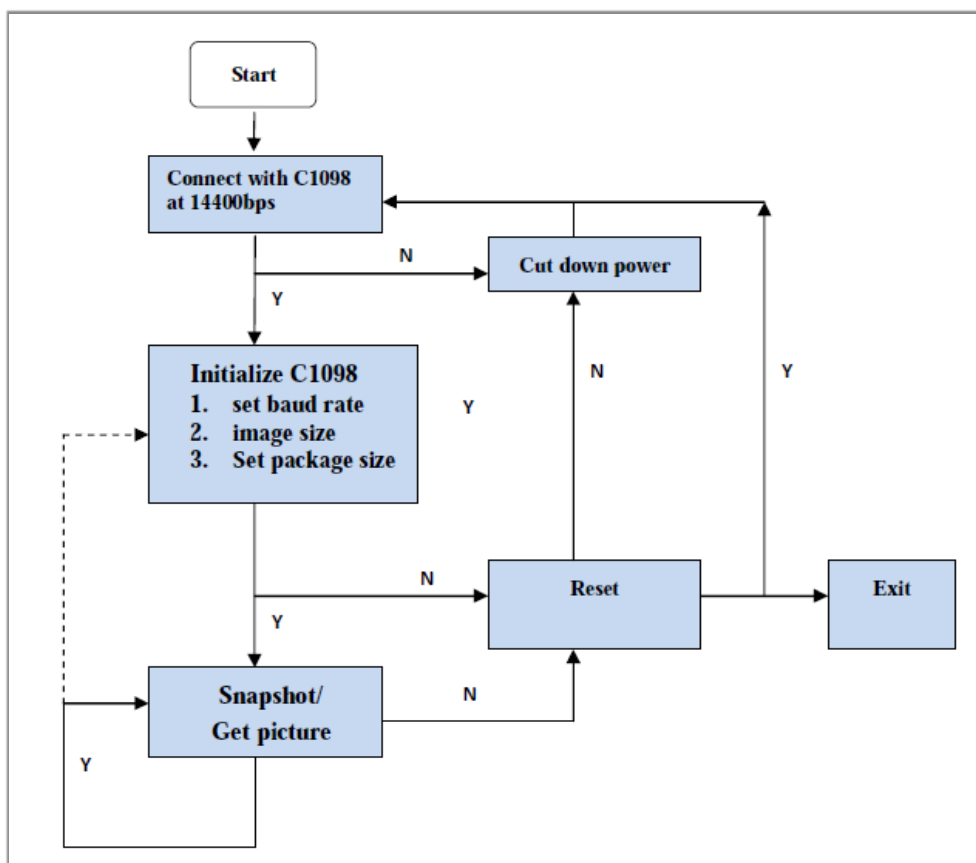


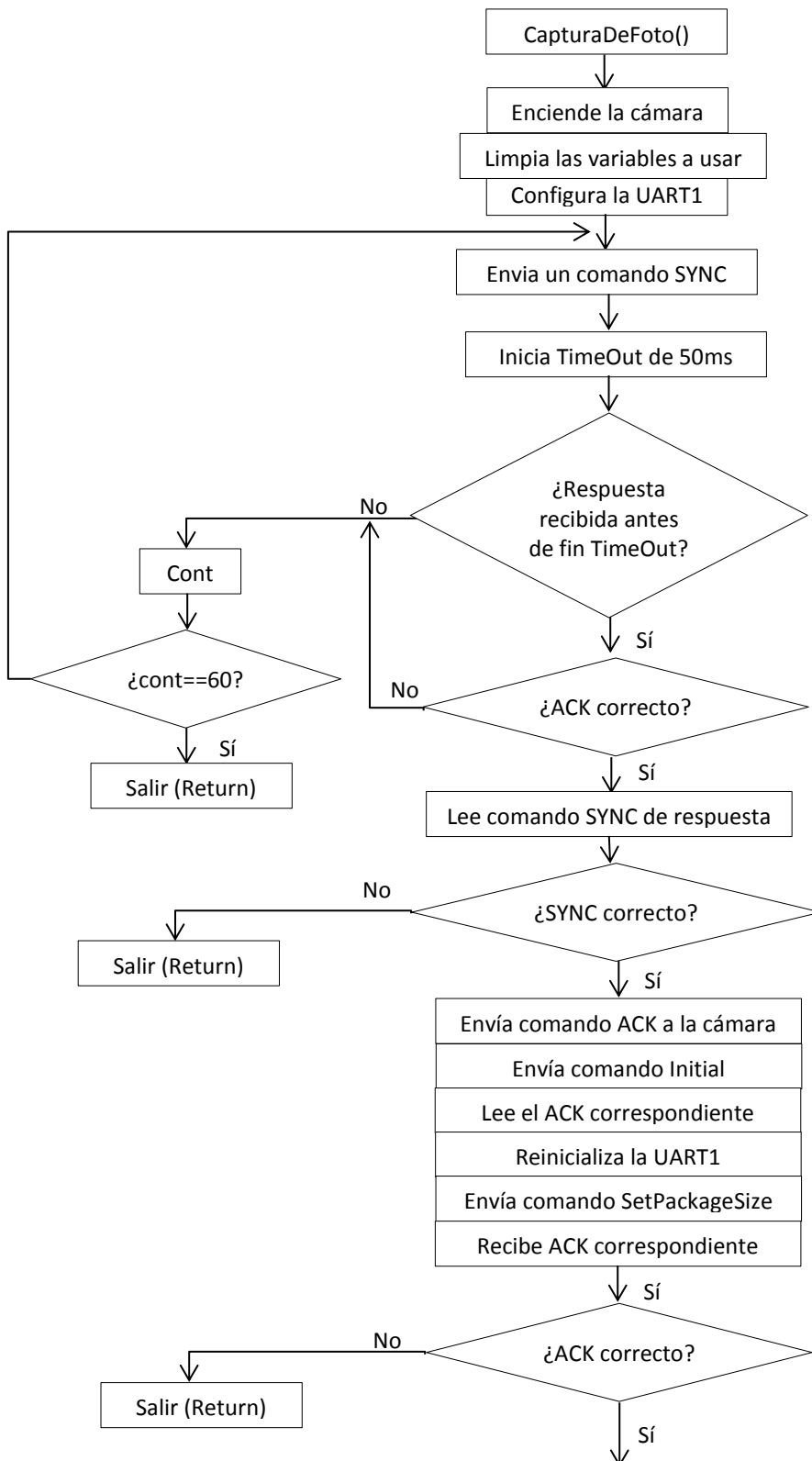
Figura 23. Diagrama de flujo proporcionado por el fabricante para comunicarse con el módulo C1098.

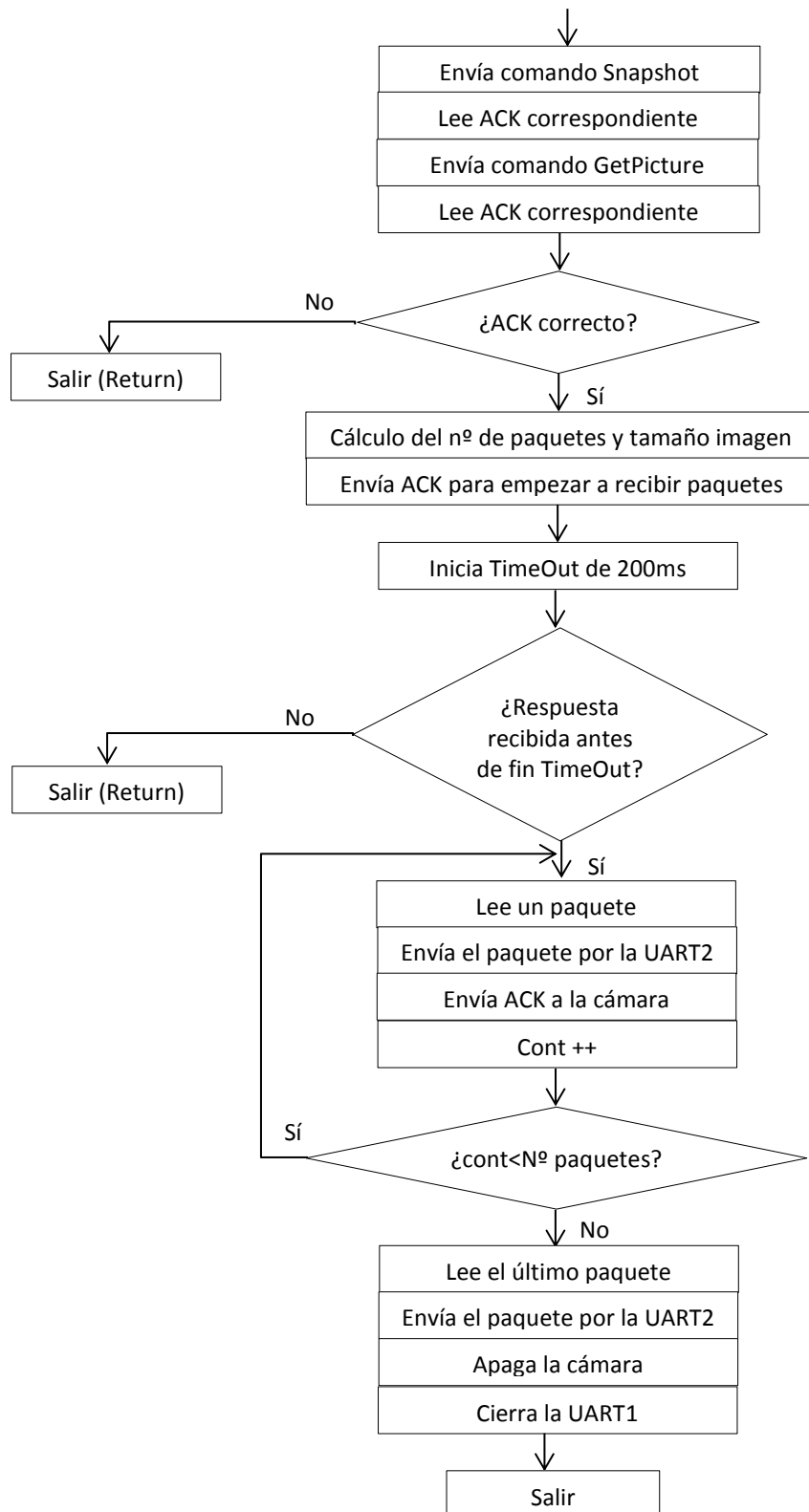
La diferencia entre este protocolo de comunicación y el realizado es que, si los comandos de asentimiento no son los correctos o si la cámara no responde, no se procede a resetear y a sincronizar de nuevo, sino que se sale directamente de la función de captura de foto. Esto se realiza así para que el proceso de captura de una foto no se alargue demasiado si la cámara no responde correctamente. Además, cada vez que se quiere capturar una foto, primero hay que encender el módulo con el control explicado en el apartado anterior, ya que, en reposo, la cámara se encuentra apagada.

Para comprobar la funcionalidad del programa, se ha utilizado TeraTerm. Para ello, se ha usado un cable DB9 conectado, por un extremo, a uno de los puertos RS-232 del ordenador y, por el otro extremo, al puerto RS-232 de la Explorer 16 que se encuentra conectado a la UART2 del microcontrolador a través de un integrado que cambia los niveles de tensión TTL a niveles RS-232.

Con el programa TeraTerm se ha creado un fichero binario con la extensión JPG, que se llenará con los datos que se reciban de la imagen. Para realizarlo, el programa realizado de captura de foto envía los datos por la UART2 cada vez que se recibe un paquete, eliminando cabeceras y finales de trama. Finalmente, se cierra el fichero JPEG creado y ya estará disponible para su visualización.

A continuación se muestra el diagrama de flujo correspondiente al programa de captura de foto realizado.





## 5. MÓDULO WIFI RN-171

El módulo RN-171 es un módulo embebido que permite la conectividad wifi. Los datos se envían y reciben por una interfaz UART que lleva integrada. Se alimenta a 3,3V y se conectará a la UART2 del microcontrolador. Al igual que en el caso de la cámara de fotos, hay que tener en cuenta que el terminal de transmisión de uno, hay que conectarlo con el de recepción del otro y viceversa.

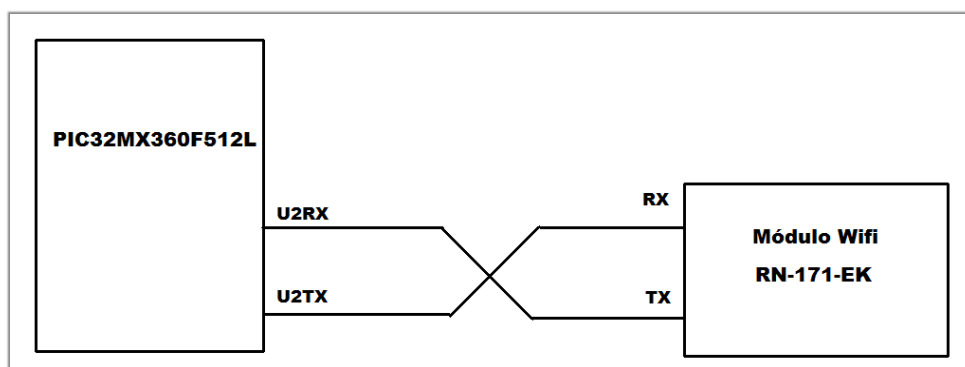


Figura 24. Esquema de la conexión del módulo wifi con el PIC32MX360F512L.

No se utilizará el módulo directamente, sino que se usará un kit de evaluación, el RN-171-EK. Esta tarjeta lleva integrada entre sus pistas la antena necesaria para la comunicación remota, además de un conector mini USB para poder comunicarse de una manera sencilla con un ordenador. La comunicación a través de la UART se puede hacer por medio del conector USB o por los terminales que integra la placa, pero no por ambos al mismo tiempo.

El módulo wifi puede operar en dos modos: el modo de datos, que es el de defecto, y el modo comandos. En el modo de datos recibe datos por wifi, quita las cabeceras TCP/IP y envía los datos por la UART. También puede transmitir los datos que reciba por la UART, construyendo un paquete TCP/IP y enviándolo por wifi. En el modo comandos se pueden realizar configuraciones al módulo, como cambiar la velocidad de transmisión, que por defecto es 9600bps. Los comandos de configuración se envían en código ASCII.

Para entrar en modo comandos se utiliza, por defecto “\$\$\$” y el módulo responderá con “CMD”.

El modo de funcionamiento del módulo wifi que se utilizará es el de conexión automática con un punto de acceso (router wifi). Para ello, el fabricante nos proporciona en la hoja de características los siguientes comandos de configuración. Una vez que esté configurado con estos comandos, el módulo se conectará a la red seleccionada automáticamente aunque se reinicie el módulo o la red wifi.

- \$\$\$ → Entra en el modo de comandos
- scan → Escanea las redes wifi que detecta en su alcance.



- `set wlan ssid NombreRed` → Se conecta a la red de nombre “NombreRed”
- `set wlan pass contraseña` → Introduce la contraseña correspondiente
- `set wlan join 1` → Se conecta automáticamente a la red
- `save` → Guarda la configuración en el archivo por defecto
- `reboot` → Fuerza al módulo a reiniciar con la configuración guardada

Para comprobar el funcionamiento de esta configuración, se ha utilizado el programa TeraTerm para enviar los comandos al puerto al que esté conectado el cable USB que se une al módulo. Cuando el módulo se encuentra conectado a una red, el LED verde parpadea despacio. Cuando un dispositivo, a través de la red, se conecta al módulo o al revés, el LED verde se encuentra encendido continuamente. En la siguiente tabla se puede ver los diferentes estados del módulo que se indican con los cuatro LEDs que integra.

Iluminación	LED Azul	LED Rojo	LED Naranja	LED Verde
Continuo	No usado	-	-	Conectado sobre TCP
Parpadeo rápido	No usado	No asociado a red	Transferencia de datos (RX/TX)	Sin dirección IP
Parpadeo lento	No usado	Asociado, sin internet	-	Con dirección IP
Apagado	No usado	Asociado, con internet	-	-

Tabla 1. Estado del módulo wifi en función de los LEDs que integra.

Una vez conectada a una red, con el programa Docklight se ha comprobado la conexión con un dispositivo remoto sobre TCP y la transmisión de datos, tanto si es el módulo el que inicia la conexión (cliente), como si es el que escucha (servidor).

En la hoja de comandos del módulo se encuentran todos los comandos disponibles para su configuración. Además de los comandos de conexión a red anteriores, se utilizarán para el proyecto los siguientes comandos:

- `set uart baudrate 115200` → Establece la velocidad de transmisión a 115200bps
- `set comm remote HELLO<EOF0>` → Establece el string que enviará el módulo cuando se conecte a un dispositivo como “HELLO<EOF0>”.

Estos comandos se enviarán junto con los de conexión a red, antes de guardar la configuración con el comando “save”.

Por último, también se utilizara el comando: `open DirecciónIPservidor NúmDePuerto`. Con este comando se hará que el módulo se conecte con el servidor.

Se han realizado pruebas para conectar el módulo wifi con un dispositivo que hacía de servidor a través de internet. Para ello, se ha utilizado el programa docklight como servidor y se ha abierto el puerto por el que escuchaba este servidor en el router wifi y también se ha configurado el router para que reenvíe los datos que le llegan por ese puerto hacia el ordenador en el que se encuentra ejecutándose docklight. También se ha desactivado el Firewall de Windows para que no de problemas en la comunicación. A continuación se ha buscado la dirección IP pública con la que el router tenía acceso a internet. Para realizar la conexión se ha conectado el módulo wifi a otra red local y se ha enviado el comando de abrir una conexión al módulo a través de TeraTerm, pero las conexiones eran fallidas. Solamente un porcentaje muy bajo de los intentos de conexión se realizaban correctamente. Pero, como se verá más adelante, interesa que casi todos los intentos de conexión que realiza el módulo con el servidor sean correctos. Por ello, el funcionamiento de este termostato queda limitado a una red local wifi, donde las conexiones entre el módulo wifi y el servidor sí que se realizan satisfactoriamente en un porcentaje alto. Sin embargo, la aplicación del usuario sí que podrá realizar conexiones con el servidor a través de internet.

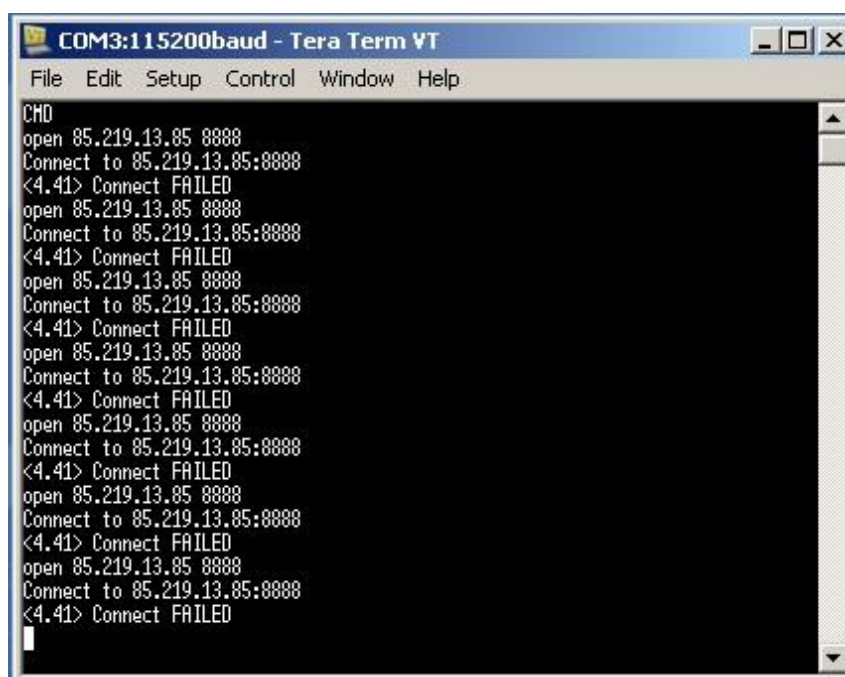


Figura 25. Intentos de conexión con un servidor a través de internet

## 5.1 APLICACIÓN DE CONFIGURACIÓN DEL MÓDULO WIFI

Cada usuario del termostato tendrá que configurar el módulo wifi para que se conecte a la red wifi de su vivienda. Hasta ahora se ha explicado cómo conectarse a una red desde un programa que interactúa con los puertos del ordenador, en el que hay que escribir todos los comandos de configuración necesarios. Sin embargo, no es algo práctico de usar, ni sencillo para todas las personas. Por ello, se ha desarrollado una aplicación de

escritorio para Windows con la que se podrá configurar el módulo para que se conecte a la red wifi del usuario de manera sencilla y visual.

El entorno de programación donde se ha desarrollado la aplicación es Visual Studio. El primer paso es realizar el diseño del interfaz de usuario, es decir, la distribución de todos los elementos que compondrán la ventana de interacción con el usuario. En la siguiente figura se puede ver el diseño final de la aplicación de configuración del módulo wifi conectado al termostato.

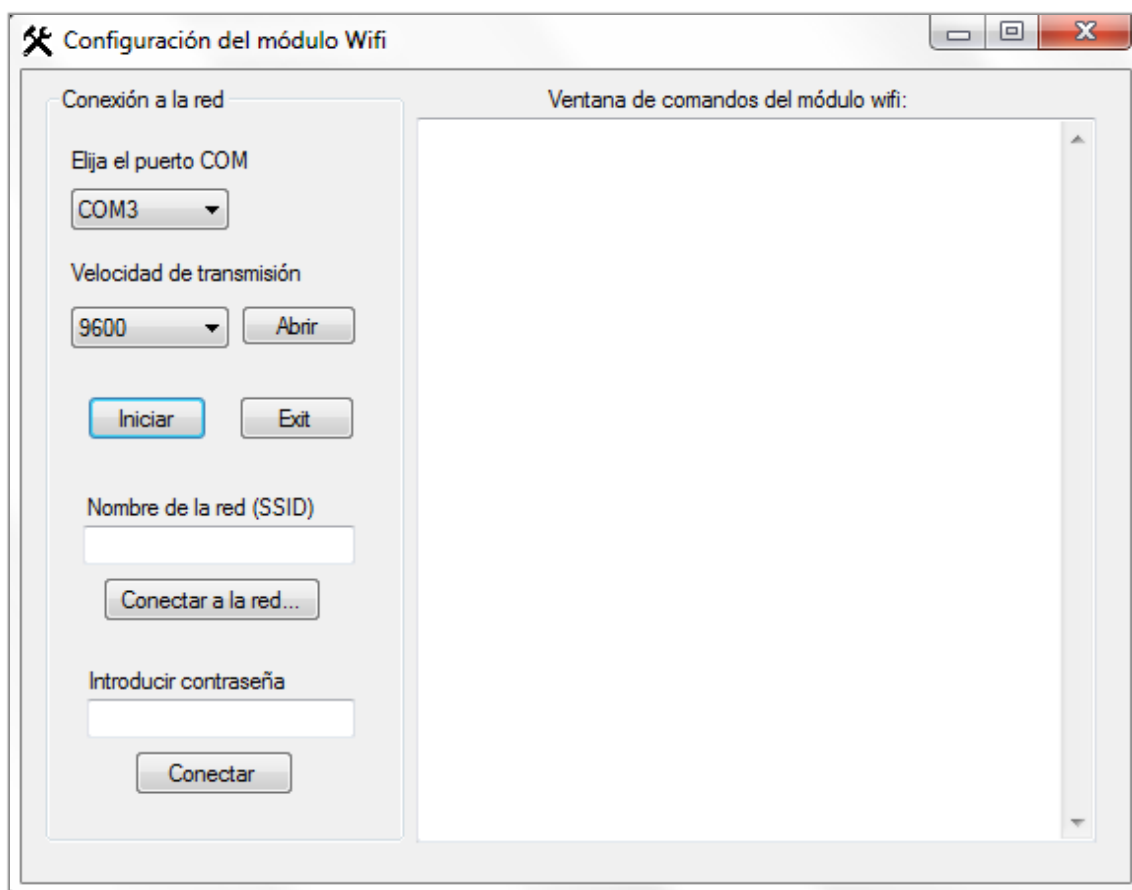


Figura 26. Interfaz de usuario de la aplicación de configuración del módulo wifi.

Después de realizar la distribución de los elementos de la interfaz de usuario, hay que desarrollar el programa que se ejecuta al interactuar con los diferentes elementos.

El nombre del puerto se elige desde un ComboBox, al igual que la velocidad de transmisión del puerto seleccionado. Como el módulo wifi trae de fábrica la velocidad de 9600bps y con la aplicación de configuración se le establece a 115200bps, sólo se podrán elegir estas dos velocidades de transmisión.

Al pulsar el botón “Abrir”, se abre el puerto seleccionado a la velocidad de transmisión seleccionada. Si el puerto no se puede abrir, o se produce algún tipo de error al abrir el puerto, se abre una ventana de mensajes indicando que se ha producido un error.

Con el botón iniciar se reestablece de fábrica al módulo para eliminar toda la configuración que lleve guardada, entra en el modo comandos y busca las redes wifi que se encuentren a su alcance.

El botón “Exit” sirve para que el módulo wifi salga del modo comandos y se vuelva a iniciar si se desea.

En el cuadro de texto donde se lee “Nombre de la red (SSID)” sirve para escribir el nombre de la red a la que se quiere conectar. Con el botón “Conectar a la red...” se establece, el nombre introducido en el cuadro de texto, como la red a la que se va a conectar.

Finalmente, se escribe la contraseña de la red wifi en el cuadro de texto donde se lee “Introducir contraseña”. Si la red no tiene una contraseña de acceso, este cuadro de texto se debe dejar en blanco. Al pulsar el botón “Conectar” se establece la conexión con la red wifi y se configura para que se conecte de forma automática si pierde la conexión o si se apaga y se vuelve a encender. Además, se cambia la velocidad de transmisión de 9600bps a 115200bps.

Desde la ventana de comandos, se puede ver toda la comunicación con el módulo wifi. Hay que tener en cuenta que al cambiar la velocidad de transmisión los mensajes que se envíen o reciban del módulo estarán malinterpretados si no se cambia también la velocidad del puerto.

En el programa, cada vez que se escribe un comando al módulo wifi desde el puerto, se inicia un timer que interrumpe a los 500ms para enviar el siguiente comando. Este tiempo de espera se establece para dar tiempo a que el módulo responda y la respuesta sea recibida por la aplicación. Además, si se pulsa cualquier botón antes de abrir el puerto, se abrirá una ventana de mensajes informando del error.

## 6. COMUNICACIÓN TERMOSTATO-APLICACIÓN.

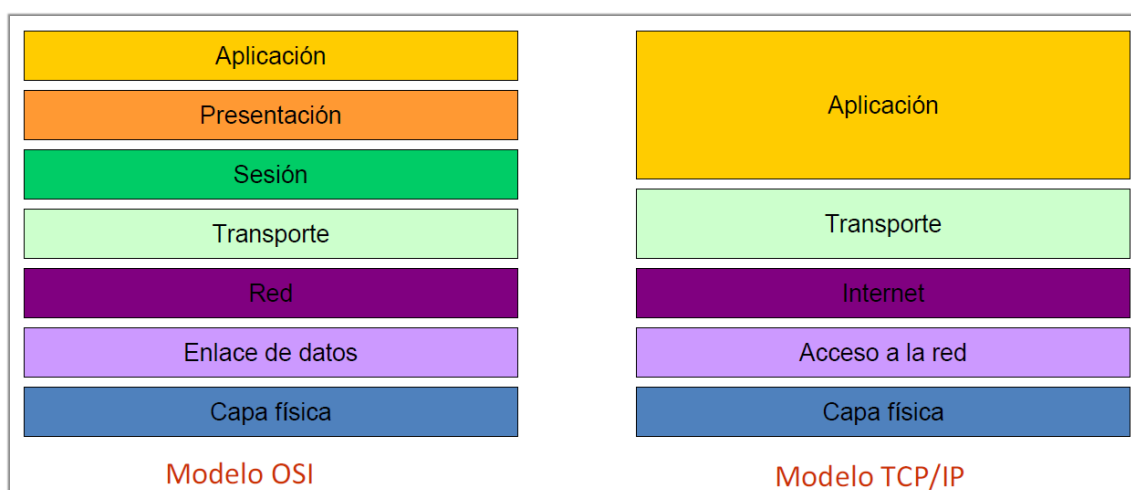
El termostato se podrá controlar remotamente desde una aplicación de escritorio vía wifi. Para ello, se necesita crear una conexión inalámbrica entre el termostato, a través del módulo wifi y la aplicación de escritorio que se diseñe.

### 6.1 CONCEPTOS PREVIOS

#### 6.1.1 MODELO TCP/IP

Para la comunicación a través de una red de ordenadores existen dos estándares que han predominado, el modelo OSI (Open System Interconnection) y el modelo TCP/IP (Transport Control Protocol/Internet Protocol). De entre ambas, ha predominado TCP/IP.

Estos modelos siguen un diseño por capas, pudiéndose hablar de una arquitectura o pila de protocolos. Cada capa utiliza servicios de la capa inferior y ofrece servicios a la capa superior.



*Figura 27. Relación entre la pila de protocolos del modelo OSI y el modelo TCP/IP.*

La capa física del modelo TCP/IP define la interfaz física y el medio de transmisión. Además, especifica el tipo de señal a transmitir, la velocidad de datos, etc.

La capa de acceso a red se encarga del intercambio de datos entre el dispositivo y la red. El emisor proporciona a la red la dirección de destino para encaminar los datos hacia el receptor.

La capa de red (Internet) encamina los datos cuando el origen y el destino se encuentran en redes diferentes. Utiliza el protocolo de internet (IP), el cual no es orientado a conexión ni confiable.

La capa de transporte ofrece dos protocolos. Uno es el TCP, orientado a conexión y confiable, ya que se entregan los datos en el destino sin errores. El otro es el UDP (User Datagram Protocol), que no es orientado a conexión ni confiable.

La capa de aplicación contiene todos los protocolos de aplicación como navegadores, correo electrónico, transferencia de ficheros, etc.

### **6.1.2 ARQUITECTURA CLIENTE-SERVIDOR**

En el modelo TCP/IP las comunicaciones entre dispositivos se basan en la arquitectura cliente servidor.

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Básicamente, un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre un solo dispositivo, es más ventajosa en un sistema multiusuario distribuido a través de una red.

El cliente es la aplicación que inicia la conexión o intercambio de datos con el servidor y la arranca el usuario cuando desea utilizar un servicio de red. Un ejemplo de cliente es un navegador web o un cliente de e-mail.

El servidor es la aplicación que recibe y acepta la solicitud de conexión o intercambio de datos con el cliente. Normalmente se ejecuta de forma continua y espera las solicitudes de los clientes. Un ejemplo es un servidor web o un servidor de e-mail.

El cliente administra la interfaz de usuario ya que es el que interactúa con éste y un servidor es el que interactúa con los servicios compartidos, actúa como motor de software que maneja recursos compartidos tales como bases de datos, impresoras, módems, etc.

### **6.1.3 SOCKETS**

Los sockets son mecanismos de comunicación bidireccional entre procesos que se ejecutan en un mismo dispositivo o en diferentes. Los sockets se pueden ver como una interfaz con la capa de transporte del modelo TCP/IP. Los sockets se usan como hilo de comunicación entre un cliente y un servidor.

Los parámetros de un socket son los siguientes:

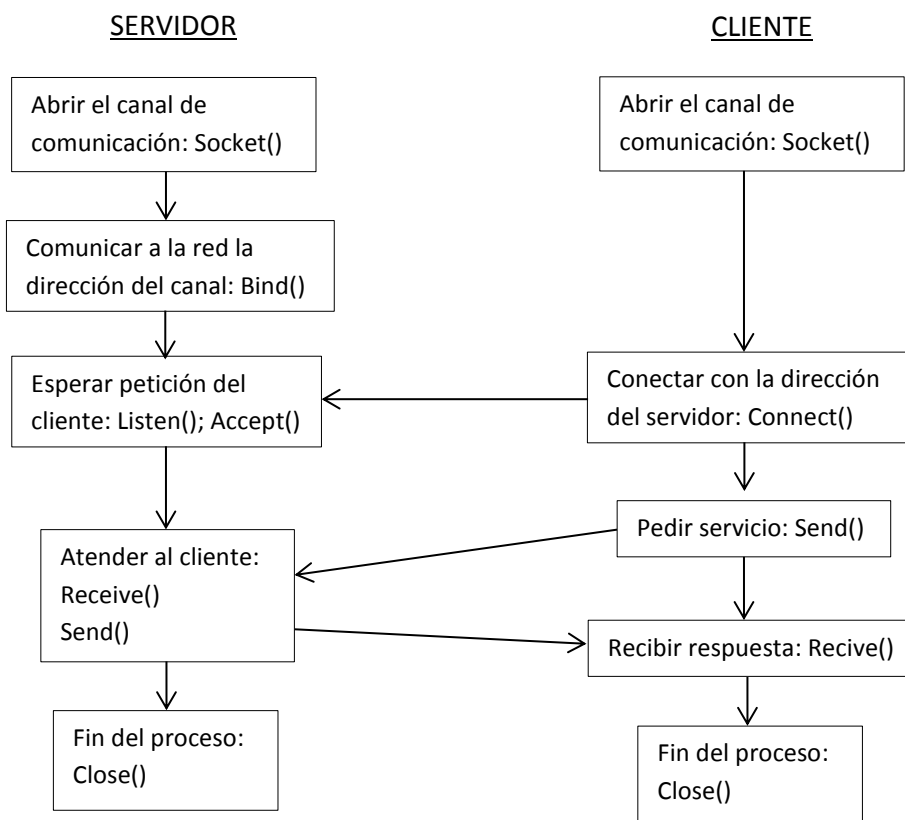
- Dirección IP del cliente.
- Puerto del cliente por el que inicia la conexión.

- Dirección IP del servidor.
- Puerto del servidor por el que escucha una conexión.

Un socket se caracteriza por tres atributos: dominio, protocolo y tipo. El dominio especifica el medio de comunicación de la red que el socket utilizará. El protocolo especifica el protocolo a utilizar. En cuanto al tipo, existen varios, los dos más empleados son el socket de flujo, orientado a conexión, y el socket de datagrama, no orientado a conexión.

En los sockets orientados a conexión, primero hay que establecer correctamente la conexión. Para gestionar la conexión se usa el protocolo TCP del modelo TCP/IP. Se garantiza que todos los datos van a llegar de un programa al otro correctamente. Para poder realizar la conexión entre cliente y servidor es necesario conocer la dirección IP del servidor y el puerto por el que escucha una conexión.

A continuación se muestran y explican los diagramas de flujo que siguen un servidor y un cliente para la comunicación mediante sockets.



En el lado del servidor, primero se crea el socket y, a continuación, se avisa al sistema operativo de que se ha abierto un socket y se quiere asociar el programa servidor a dicho socket. Después se avisa al sistema de que comience a atender dicha conexión de red. Luego se empieza a aceptar las conexiones de los clientes. Si no hay clientes se quedará bloqueada hasta que algún cliente se conecte. Cuando se produce una conexión se recibe y se escriben los datos. Tanto el cliente como el servidor deben saber qué datos esperan recibir y en qué formato. Finalmente, se cierra el socket.

En el lado del cliente, primero se crea el socket, al igual que en lado del servidor. A continuación, se solicita la conexión con el servidor. Esta función se quedará bloqueada hasta que el servidor acepte la conexión. En esta solicitud se debe proporcionar la dirección IP del servidor y el número de puerto que se desea. Después, se escriben y reciben los datos y, finalmente, se cierra la comunicación.

#### **6.1.4 CONCEPTOS DEL WI-FI**

El wifi es un mecanismo de conexión de dispositivos de manera inalámbrica. Wifi es una marca de la Wi-Fi Alliance, la organización comercial que adopta, certifica y prueba que los equipos cumplen los estándares 802.11 de IEEE (Institute of Electrical and Electronics Engineers).

En un principio, era habitual que los puntos de acceso inalámbricos fuesen de un fabricante y los visitantes que intentaban comunicarse con ellos no pudiesen, pues tenían hardware de otros fabricantes. Por esta razón, la Wi-Fi Alliance estandarizó el mercado con la marca WiFi.

Existen diversos tipos de wifi, cada uno de ellos basado en un estándar IEEE 802.11. Los estándares IEEE 802.11b, IEEE 802.11g e IEEE 802.11n disfrutaban de una aceptación internacional debido a que la banda de 2,4GHz está disponible casi universalmente, con una velocidad de hasta 11Mbps, 54Mbps y 600Mbps respectivamente.

La red wifi de una vivienda es una WLAN (Wireless Local Area Network). El router hace de punto de acceso a internet y permite la comunicación con otras WLAN.



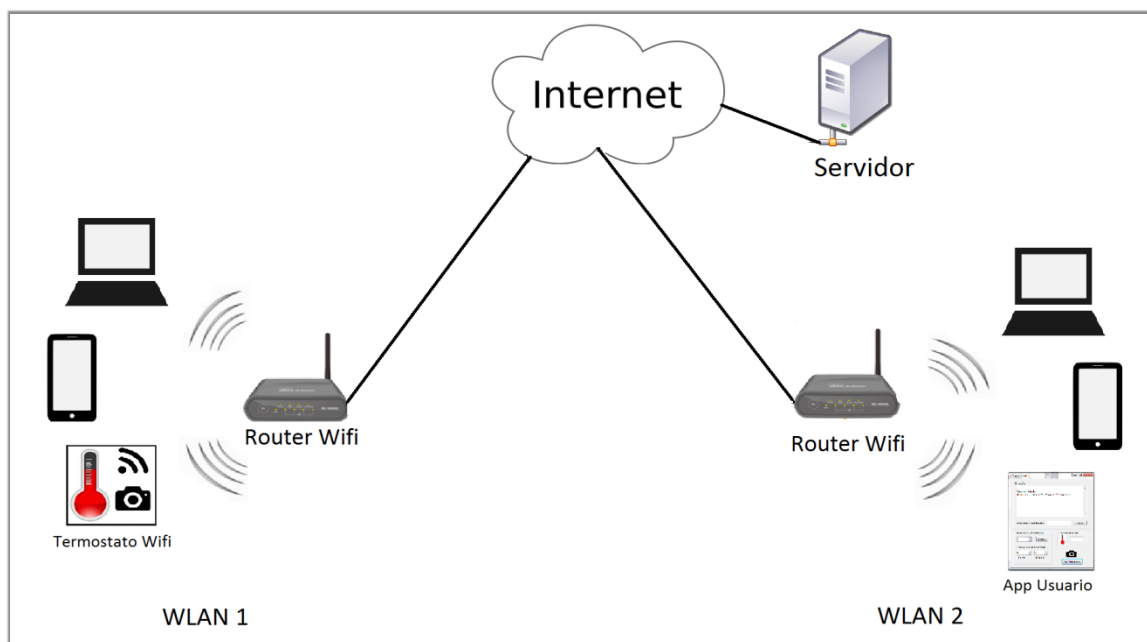


Figura 28. Disposición de redes WLAN conectadas a internet

### 6.1.5 DIRECCION IP

Una dirección IP es una etiqueta numérica que identifica, de manera lógica y jerárquica, a una interfaz de un dispositivo dentro de una red que utilice el protocolo IP. Las direcciones IPv4 se expresan en un número binario de 32 bits, permitiendo una cantidad de direcciones de  $2^{32}$ . La forma habitual de representarla es separándola en cuatro octetos y en forma decimal, que podrá tomar valores entre 0 y 255. Cada octeto se separa por un punto. Ejemplo: 80.204.116.93

Debido al incremento de dispositivos que se conectan a internet, no es suficiente con la cantidad de direcciones IP que permite la IPv4, por lo que se creó las direcciones IPv6, que están compuestas por 128 bits y se expresa en una notación hexadecimal de 32 dígitos.

Se debe de distinguir entre IP privada e IP pública. La IP privada se utiliza para identificar equipos o dispositivos dentro de una red local, como puede ser la red wifi de una vivienda. Sirven para dar conectividad entre equipos internos sin que se pueda acceder a internet. En cambio, una dirección IP pública es la que se le asigna a un dispositivo que se conecta de forma directa a internet. Son siempre únicas, no se pueden repetir. Esto permite que dos equipos con estas direcciones se puedan conectar a través de internet.

También hay que hacer una distinción entre IP fija e IP dinámica. La IP fija, como su nombre indica, no cambia. Si es una IP privada siempre se le asignará la misma IP privada dentro de la red local. Si es un IP pública, siempre se le asignará la misma dirección cuando

acceda a internet. Son comunes en proveedores de telefonía, servidores, servicios web, etc. Sin embargo, una IP dinámica es una dirección variable. Un dispositivo puede tener una IP en cierto momento y una distinta en otro. Muchos proveedores de internet asignan IPs dinámicas a sus clientes. Si uno de ellos se desconecta de internet, el proveedor puede reasignar esa dirección a otro cliente que se conecte. Normalmente, tener una dirección IP fija para acceder a internet tiene un sobre coste.

## **6.2 ESTRUCTURA DE LA COMUNICACIÓN TERMOSTATO-SERVIDOR-USUARIO. SOLUCIÓN AL PROBLEMA DE IP DINÁMICA.**

A la hora de comunicar el termostato con la aplicación del usuario a través de internet hay que tener en cuenta ciertas consideraciones. La aplicación de usuario siempre va a actuar como cliente, porque será siempre el que inicie la conexión cuando el usuario lo desee. Por tanto, el termostato tendría que hacer de servidor. Debido a que al conectarse a internet, el termostato tendrá un IP dinámica, no habrá forma de que la aplicación cliente conozca la dirección IP pública del termostato, así que no podrá conectarse con éste directamente. Además, el router wifi al que esté conectado el termostato tendría que tener el puerto por el que el termostato escucha las conexiones abierto y también redirigir los datos que lleguen por ese puerto a la IP privada del termostato. Esta configuración del router wifi lo tendría que hacer cada usuario que utilice este termostato en su vivienda, cosa que no es factible.

Por tanto, es necesario un tercero, un servidor que tenga una dirección IP pública fija. Conociendo la dirección IP fija del servidor y el puerto por el que escucha se puede establecer una conexión desde la aplicación del usuario, o desde el termostato, que ahora haría también de cliente.

La primera idea planteada consistía con que el termostato se conecte cada cierto tiempo con el servidor (el resto de tiempo estaría escuchando por un puerto a una conexión). El servidor guardaría la dirección IP pública dinámica con la que se ha conectado el termostato y cerraría la conexión. Cuando el usuario se conecte desde su aplicación, el servidor le enviaría la última dirección IP pública dinámica del termostato que haya recibido. Conociendo esta dirección que tendría el termostato en ese momento, el usuario se podría conectar directamente con el termostato y controlarlo en tiempo real. Con esta configuración se soluciona el problema de la IP dinámica, pero no el problema de tener que abrir el puerto por el que escucha el termostato y hacer que el router redirija la información que llega por ese puerto hacia el termostato.

La solución final que se ha adoptado es que el termostato envíe información de su estado al servidor cada cierto tiempo. El servidor actualizará la información que tenga de ese termostato y le enviará los posibles cambios que el usuario haya solicitado hacer en el

termostato. Cuando el usuario se conecte con el servidor desde su aplicación, el servidor le enviará la última información que haya recibido del termostato y recibirá los cambios que el usuario solicite que se haga en el termostato. Estos cambios solicitados serán enviados por el servidor al termostato la siguiente vez que se conecte. Con esta configuración quedaría resuelto el problema de IP dinámica y no sería necesario que el usuario del termostato tenga que hacer cambios en la configuración de su router wifi.

En la siguiente imagen se puede observar un diagrama con las comunicaciones que se producirán entre termostato-servidor y entre aplicación de usuario-servidor.

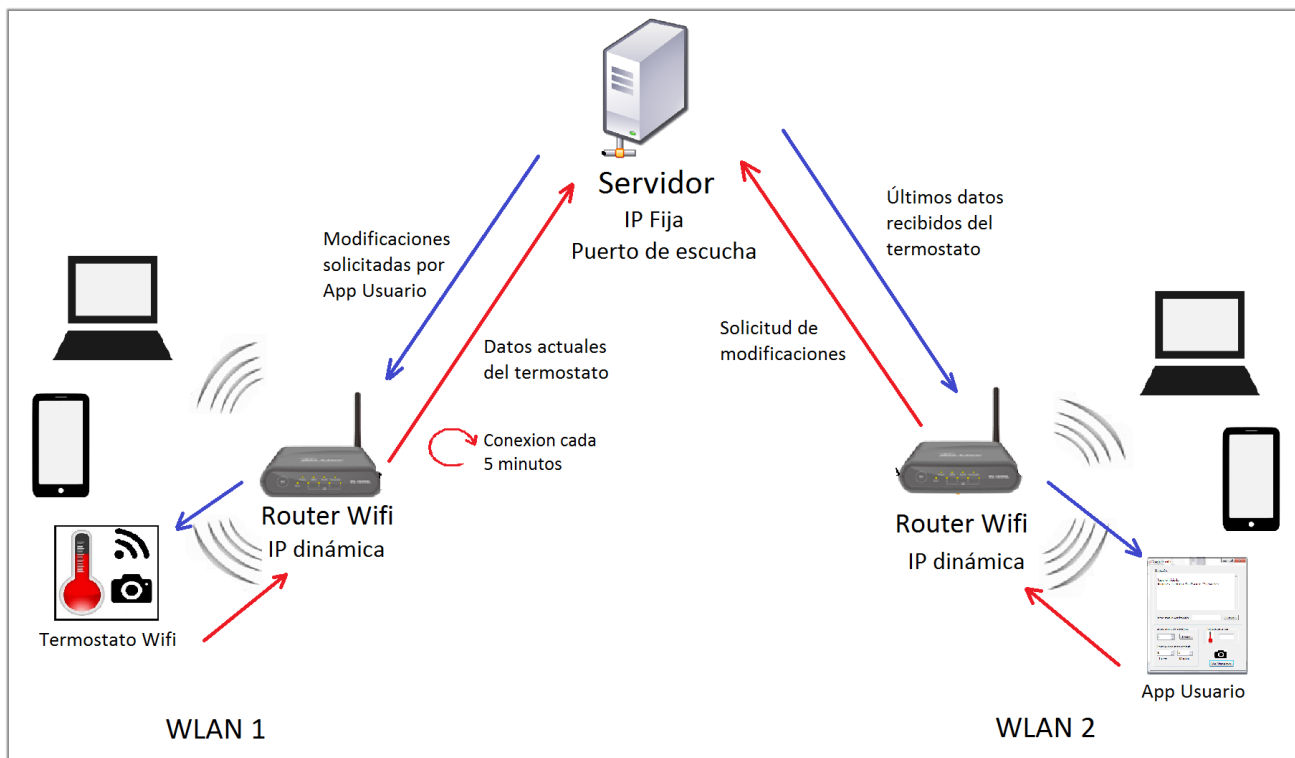


Figura 29. Diagrama de las conexiones termostato-servidor-usuario.

### 6.3 PROTOCOLO DE COMUNICACIÓN ENTRE TERMOSTATO-SERVIDOR-APLICACIÓN DE USUARIO.

El programa del servidor deberá ser capaz de dar soporte a varios termostatos y a varios usuarios desde su aplicación. Siendo capaz de gestionar correctamente la información que reciba de cada uno de ellos. En este proyecto, el servidor dará soporte a 5 termostatos diferentes para simular esta capacidad de multi-gestión.

### 6.3.1 TRAMAS DE DATOS TERMOSTATO-SERVIDOR

El protocolo de comunicación entre un termostato y el servidor consiste en la conexión entre ambos cada 5 minutos para el intercambio de datos. El termostato es el que inicia la conexión y lo hace a través del módulo wifi. Para abrir la conexión se le envía el siguiente comando por la UART al módulo wifi:

- open *DirecciónIPservidor NúmDePuerto*

Con este comando, el módulo se conecta al servidor y le envía el string que se le habrá establecido para que envíe cada vez que se conecte a un dispositivo. En este caso será: "HELLO<EOF0>".

Cuando la conexión esté establecida, el termostato enviará al servidor la siguiente trama de datos:

```
HELLO<EOF0>MODULO:NúmeroID<EOF1>password<EOF2>TemperaturaReal<EOF3>  
TemperaturaReferencia<EOF4>TiempoInicioHoras<EOF5>TiempoInicioMinutos<EOF6>  
DatosFoto<EOF7>
```

Cada <EOFx> es un final de dato, para que el programa de servidor sea capaz de interpretar y hacer lo que corresponda con cada dato. "NúmeroID" es un número de identificación que se le asigna a cada termostato para que el programa servidor pueda identificar qué termostato se ha conectado y pueda guardar la información donde corresponda. "password" es la contraseña que el usuario deberá introducir en su aplicación para poder acceder a los datos de su termostato. "TemperaturaReal" y "TemperaturaReferencia" son la temperatura actual medida por el sensor y la temperatura actual de referencia del termostato, respectivamente. "TiempoInicioHoras" y "TiempoInicioMinutos" es el tiempo restante para que la temperatura de referencia cambie a la que el usuario haya asignado desde su aplicación. Este tiempo lo asigna el usuario desde su aplicación. "DatosFoto" son todos los datos de la imagen que captura la cámara de fotos en ese momento.

Todos estos datos enviados desde el termostato están en código ASCII, menos los datos de foto, que son datos binarios. El servidor tendrá que ser capaz de interpretar correctamente todos estos datos.

Desde que el servidor reconoce que un termostato que se ha conectado, espera hasta recibir <EOF7>, que es el final de trama de datos que envía un termostato. Entonces, guarda los datos donde corresponda y procede a responder enviando la trama de datos que le corresponda a ese termostato.

Esta trama de datos que el servidor envía a un termostato tendrá la siguiente estructura:

*TemperaturaReferencia\$TiempoInicioHoras%TiempoInicioMinutos&*

Tanto \$, como % y & son finales de datos para que el termostato sea capaz de distinguir cada uno de ellos. “TemperaturaReferencia” es la temperatura de referencia que haya establecido el usuario desde su aplicación o, si no la ha modificado, será la referencia que el termostato haya enviado. “TiempoInicioHoras” y “TiempoInicioMinutos” es el tiempo restante para configurar la nueva referencia, que será la que haya introducido el usuario, o el tiempo restante que haya enviado el termostato si el usuario no lo ha modificado.

### 6.3.2 TRAMAS DE DATOS APLICACIÓN DE USUARIO - SERVIDOR

El usuario podrá acceder a su termostato desde su aplicación. La aplicación es la que hace de cliente, por lo que es la que inicia la conexión. Cuando la conexión se ha establecido, la aplicación envía una trama de datos y el servidor espera hasta que recibe el final de trama.

La estructura de la trama que envía la aplicación de usuario al servidor es la siguiente:

*USUARIO:NúmeroID<EOF1>password<EOF2>TemperaturaReferencia<EOF3>  
TiempoInicioHoras<EOF4>TiempoInicioMinutos<EOF5>*

Cada <EOFx> es un final de dato, para que el servidor pueda separar cada uno de ellos. “NúmeroID” es el número de identificación del termostato sobre el que se va a actuar. Este número permitirá al servidor asociar a la aplicación con los datos de su termostato. “password” es la contraseña que el usuario introduce en su aplicación para para que el servidor le permita el acceso a los datos de su termostato. “TemperaturaReferencia” es la temperatura de referencia que el usuario haya introducido en la aplicación. “TiempoInicioHoras” y “TiempoInicioMinutos” es el tiempo restante que el usuario haya introducido en la aplicación para configurar la nueva referencia. De esta manera, el usuario podrá seleccionar dentro de cuánto tiempo desea que se establezca la temperatura de referencia que introduzca.

Si el usuario vuelve a cambiar los datos de temperatura de referencia y/o tiempo restante, la trama que se le envíe al servidor tendrá la misma estructura siempre.

El servidor puede enviar dos tramas diferentes a la aplicación de usuario. Una para cuando la contraseña que haya enviado no se corresponda con la del termostato al que quiere acceder, y otra cuando la contraseña sea la correcta, con todos los datos del termostato.

Cuando la aplicación de usuario envía la contraseña incorrecta, el servidor le responde con la siguiente trama:

Clave incorrecta! Vuelva a introducir la clave.<EOF1>

Cuando la aplicación de usuario envía la contraseña correcta, el servidor le responde con la siguiente estructura de trama:

Clave correcta!<EOF1>TemperaturaReal<EOF2>TemperaturaReferencia<EOF3>  
TiempoInicioHoras<EOF4>TiempoInicioMinutos<EOF5>DatosFoto<EOF6>

Al igual que antes, <EOFx> son finales de datos para que la aplicación de usuario sepa interpretarlos. “TemperaturaReal” es última temperatura medida con el sensor que haya enviado el termostato al servidor. “TemperaturaReferencia” es la última temperatura de referencia que haya enviado el termostato al servidor. “TiempoInicioHoras” y “TiempoInicioMinutos” es el tiempo restante que el servidor almacena de su última conexión con el termostato, para que el termostato establezca la temperatura de referencia solicitada. Dado que este valor se actualiza cada vez que se conecta el termostato al servidor, el dato que reciba el usuario no será el tiempo restante real, pero tendrá una desviación de hasta 5 minutos, que es el tiempo que tarda el termostato en reconectarse con el servidor, por lo que no se considera un error grande tratándose de establecer una temperatura de referencia al pasar un tiempo determinado.

## 7. PROGRAMACIÓN DEL SERVIDOR

El programa del servidor se ha realizado como una aplicación de consola de Windows. El entorno de desarrollo utilizado es Visual Studio y se ha programado en C#. Para la programación se ha utilizado como referencia el programa ejemplo de socket asíncrono de servidor de la página web <https://msdn.microsoft.com/>. Desde esta página Microsoft proporciona documentación, códigos de ejemplo, información técnica, artículos y guías de referencia para desarrolladores de software con sus compiladores.

La ventaja de utilizar un socket asíncrono en lugar de uno síncrono es que la ejecución del programa del servidor no se suspende mientras espera una conexión de un cliente. Concretamente, el ejemplo de socket asíncrono de servidor se encuentra en la siguiente página:

<https://msdn.microsoft.com/es-es/library/fx6588te%28v=vs.110%29.aspx>

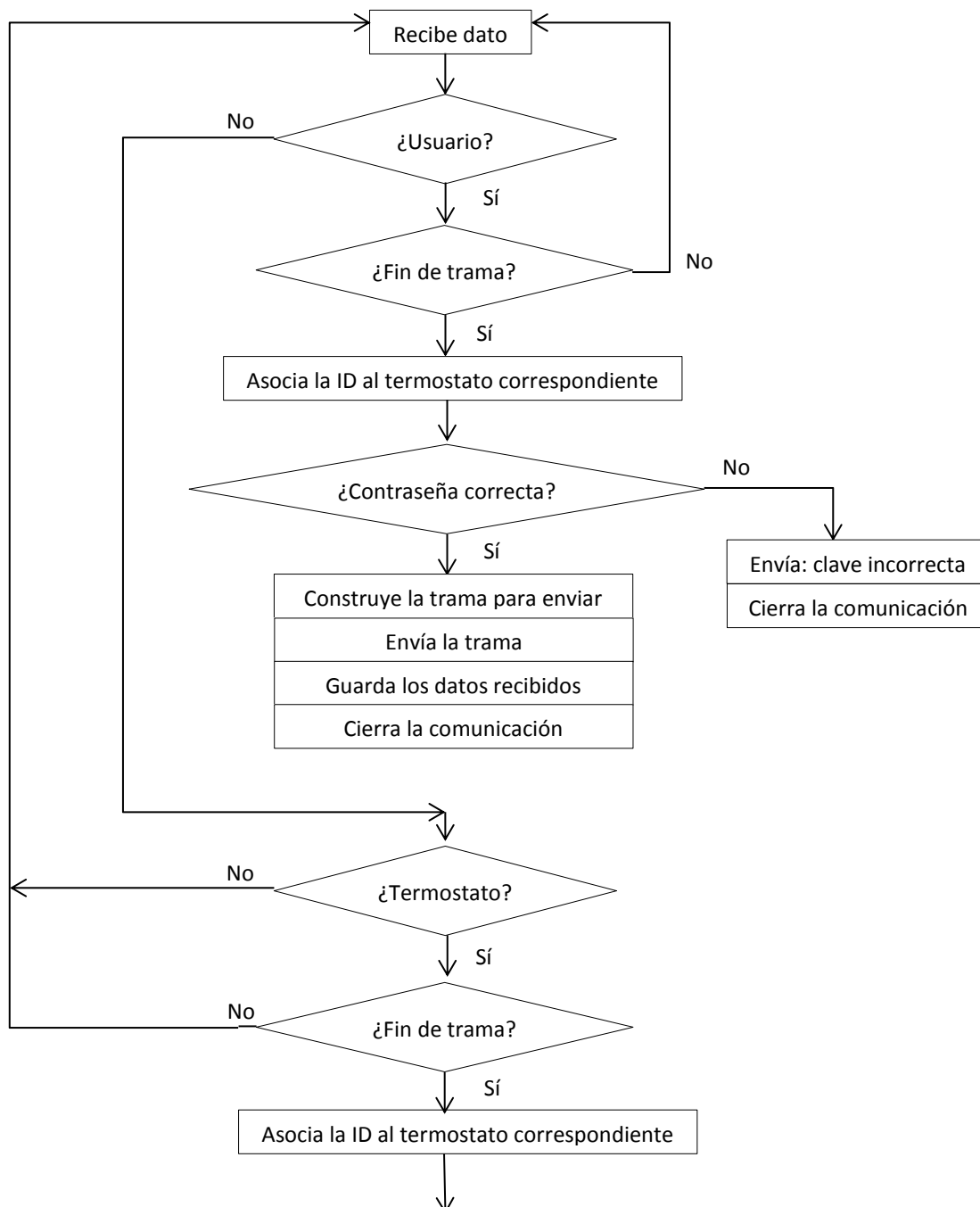
Este ejemplo ha sido modificado en gran medida para que realice las operaciones que se desean con el termostato y la aplicación del usuario. En el propio programa del servidor se encuentra la explicación detallada de cada parte, en los comentarios del mismo. Los comentarios del programa que se encuentran en inglés son las partes de programa pertenecientes al ejemplo utilizado de referencia. Las partes del programa con comentarios en castellano son de elaboración propia.

Para la recepción de datos de cliente, hay que distinguir los dos tipos de tramas que pueden llegar del cliente. El termostato wifi enviará una trama, mientras que la aplicación de usuario enviará otro tipo de trama. Cada vez que un cliente se conecta, al recibir los datos se distinguen si son del termostato, cuya trama contiene en su cabecera "MODULO", o si son de la aplicación de usuario, cuya trama contiene "USUARIO" en su cabecera. Una vez distinguida de cuál es la trama, se siguen recibiendo datos hasta que se recibe el fin de trama que corresponda, "<EOF5>" en caso de la aplicación de usuario y "<EOF7>" en caso del termostato.

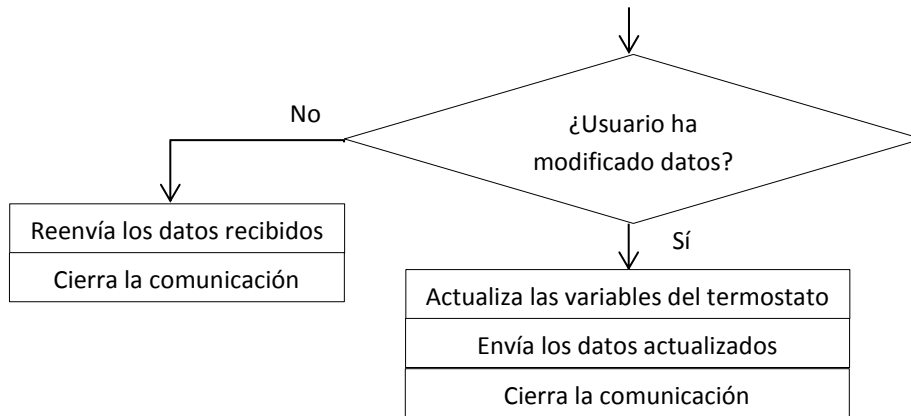
Si la trama recibida es de la aplicación de usuario, primero se compara su número de ID con el de los termostatos a los que da soporte. Una vez asociado al termostato, se comprueba si la contraseña recibida del usuario es la almacenada de ese termostato. Si no es igual, se envía el mensaje de que no es la contraseña correcta. Si es igual, se envía la trama con los datos del termostato que se ha asociado. Por último, se almacenan los datos que el usuario quiere modificar del termostato para enviárselos la próxima vez que éste establezca una conexión.

Si la trama recibida es del termostato, primero se comprueba su número ID con la de los termostatos que da soporte para asociarlo al que corresponda. A continuación se almacenan los datos recibidos en la trama y se comprueba si el usuario ha modificado los datos que se van a enviar al termostato para enviar estos datos. Si no se han modificado,

se reenvían los datos recibidos por el termostato en esta conexión. Finalmente se construye la trama para el termostato y se envía. Hay que tener en cuenta que los datos se envían y reciben como bytes de 8 bits y los que se reciben del termostato están en código ASCII (7 bits), menos los datos de la foto, que son en binario. Por ello, los datos recibidos no se codifican a ASCII (7 bits) sino que se codifican en UTF8 (8 bits), así no se pierde información (1 bit por cada byte). Si se codifica toda la trama recibida a ASCII, al obtener los datos de la foto no serían los correctos porque los archivos jpg son binarios. Los datos de la foto se obtienen de la trama recibida sin codificar, el resto de los datos se obtienen de la trama recibida codificada en UTF8.







## 8. PROGRAMACIÓN DE LA APLICACIÓN DE USUARIO

Para programar la aplicación de usuario también se ha utilizado el entorno de desarrollo Visual Studio con C#, pero esta vez será una aplicación de escritorio de Windows, en lugar de una de consola como el servidor.

El primer paso para el diseño de la aplicación de usuario es colocar y distribuir los diferentes elementos que se utilizarán en la interfaz del usuario. En la siguiente imagen se puede ver el diseño de la aplicación del usuario. El siguiente paso consiste en programar todas las funciones que se quiere que hagan los diferentes elementos de la aplicación.

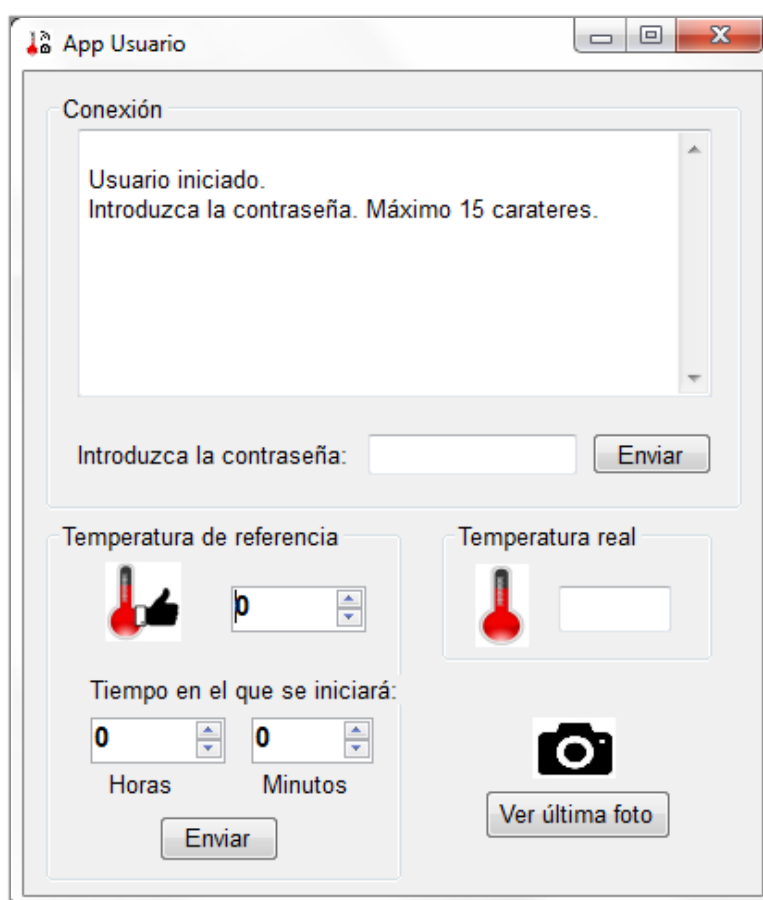


Figura 30. Interfaz de usuario de la aplicación de usuario.

Al iniciar la aplicación, se indicará, en la ventana de texto, que la aplicación se ha iniciado y que el usuario puede introducir la contraseña. En esta ventana de texto, el usuario podrá leer la actividad que se realiza. Además, se configura el cuadro de texto donde se introduce la contraseña para que no sea visible al escribirla.

Más abajo se encuentra otro cuadro de texto, donde se escribirá la contraseña que se le haya asignado al usuario con el termostato. Con esta contraseña podrá acceder en el servidor a los datos de su termostato. Al pulsar el botón “Enviar” situado junto a la contraseña, se establecerá una conexión con el servidor donde se enviará y recibirá la trama correspondiente y finalmente se cerrará la conexión. La trama que se envía solo

contiene el número ID del termostato y la contraseña, los demás campos de la trama se envían vacíos, pues solo interesa que el servidor dé acceso y envíe los datos del termostato.

Para programar esta conexión con el servidor, se ha utilizado como referencia el ejemplo de socket asíncrono de cliente de la misma página web donde se obtuvo el ejemplo de servidor: <https://msdn.microsoft.com/>.

La ventaja de utilizar un socket de cliente asíncrono es que la ejecución de la aplicación cliente no se suspende mientras el servidor devuelve la respuesta. La página donde se encuentra este ejemplo es la siguiente:

[https://msdn.microsoft.com/es-es/library/bew39x2a\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/bew39x2a(v=vs.110).aspx)

Al igual que en el caso del servidor, el programa ejemplo de partida se ha modificado en gran medida para adaptarlo a las funciones que se desea que realice la aplicación del usuario. Básicamente para enviar, recibir y gestionar las tramas que se tienen que enviar ambos. Los comentarios del programa que se encuentran en inglés son las partes de programa pertenecientes al ejemplo utilizado de referencia. Las partes del programa con comentarios en castellano son de elaboración propia. En el programa se explica cada parte del programa utilizada.

Una vez recibida la trama del servidor, se muestra en la interfaz de usuario la temperatura real, la de referencia en el momento que el termostato haya enviado los datos al servidor, y ya se podrá pulsar el botón de “Ver última foto” que abrirá directamente la foto que el termostato haya enviado al servidor por última vez.

Después de seleccionar la temperatura de referencia deseada y el tiempo en el que se quiere que se establezca esa temperatura de referencia, se pueden enviar estos datos pulsando el botón “Enviar” situado en el cuadro de la temperatura de referencia. Al pulsar este botón, se vuelve a establecer otra conexión con el servidor y se enviara la trama con los datos. Como no interesa volver a recibir los datos, la conexión que se realiza es más simple que la del ejemplo de socket de cliente asíncrono utilizada como referencia en la primera conexión.

## 9. PROGRAMACIÓN GENERAL DEL TERMOSTATO

El programa principal se ha realizado en C, en el entorno de desarrollo MPLABX. El programa cuenta con varios ficheros. Unos son ejemplos de programación para controlar la pantalla LCD y otros se han realizado como parte del proyecto.

Los ficheros ejemplo utilizados para el control de la pantalla LCD son:

- Explore.c
- explore.h
- LCD.h
- LCDlib.c

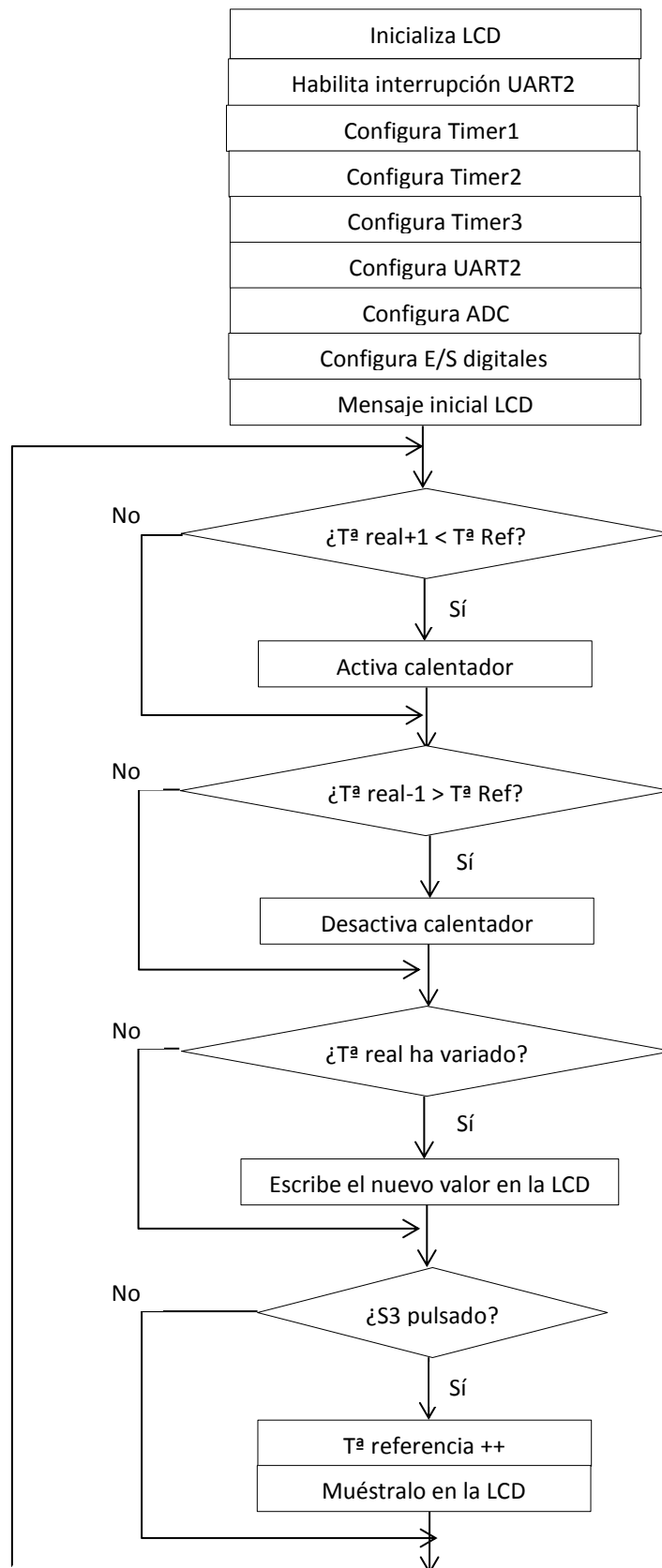
Los ficheros creados para el proyecto son:

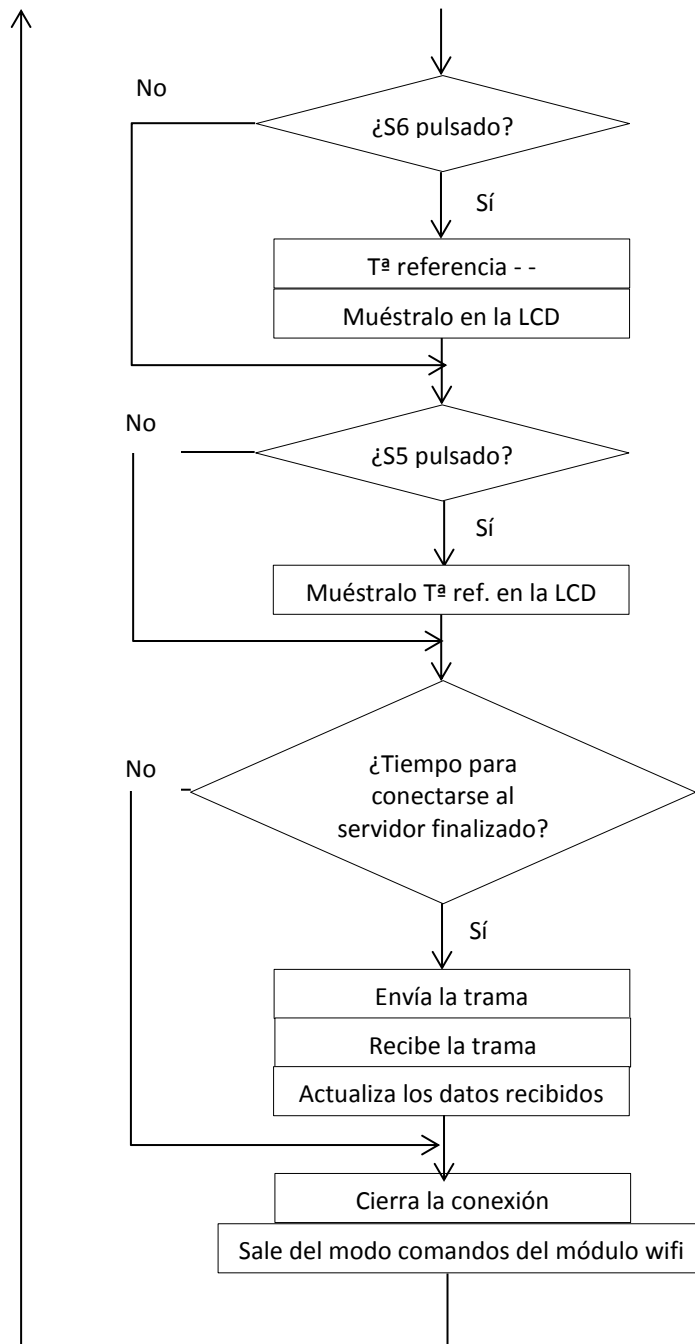
- Inicializaciones.c
- CapturaDeFoto.c
- Main Termostato.c

El fichero “Inicializaciones.c” contiene las funciones para configurar la UART1, la UART2, el Timer1, el Timer2, el Timer3 y el ADC. El fichero “CapturaDeFoto.c” incluye la programación y las funciones necesarias para la captura de foto con el módulo de la cámara, tal y como se ha explicado en el apartado 4.3. El fichero “Main Termostato.c” contiene el programa principal del termostato, además de otras funciones necesarias para el programa, como una conversión A/D, TimeOut, o las interrupciones.

El programa principal empieza con las inicializaciones y configuraciones necesarias para ejecutar el programa, y con un mensaje de inicio en la pantalla LCD. A continuación, entra en un bucle infinito para que se ejecute el programa indefinidamente. Dentro del bucle, lo primero que hay es el controlador software por histéresis del calentador, que lo encenderá o apagará según corresponda. A continuación se comprueba si la temperatura medida por el sensor ha variado, para actualizarla en la pantalla LCD. Se comprueba si ha variado porque, si se actualiza siempre, se nota un parpadeo en la pantalla LCD. Después se comprueba si el botón S3 se ha pulsado, si es así, se aumentará en uno el valor de la temperatura de referencia y se mostrará en la pantalla LCD durante 1500ms, sin que el programa se quede bloqueado. Después se comprueba si se ha pulsado el botón S6, con el que se disminuye la temperatura de referencia en un valor y se muestra en la pantalla LCD durante 1500ms sin que el programa se quede bloqueado. Seguidamente, se comprueba si se ha pulsado el botón S5, que sirve simplemente para mostrar la temperatura de referencia por la pantalla LCD sin modificarla, durante 1500ms sin bloquear el programa. Finalmente, se comprueba si el tiempo establecido para realizar una conexión con el servidor ha finalizado. Si es así, se envían los comandos al módulo wifi para que se conecte y se comprueba si la conexión se ha realizado correctamente. Si se ha conectado, envía la trama correspondiente, incluido los datos de la foto que se realizará en ese momento, cerrará la conexión y volverá al inicio del bucle infinito.

A continuación se muestra el diagrama de flujo del programa principal realizado para controlar el termostato.





Los detalles sobre la realización del programa y sobre el envío, recepción y gestión de las tramas de datos en la conexión con el servidor se encuentran incluidos en el programa como comentarios.

## 10. PROTOTIPO EN FUNCIONAMIENTO

En este apartado se mostrará el funcionamiento del prototipo con imágenes y algunas explicaciones.

Para que el sistema de comunicación wifi funcione, inicialmente el servidor tiene que estar en ejecución para dar soporte al termostato wifi y a la aplicación del usuario. Además, en el programa de la aplicación de usuario y en el del termostato, la dirección IP a la que se conectan tiene que ser la que el servidor tenga en el equipo en el que se ejecuta. En la siguiente imagen se muestra el programa del servidor en ejecución y a la espera de conexiones. En el caso de esta prueba, servidor tenía la dirección IP 192.168.43.110 y escuchaba por el puerto 8888.



*Figura 31. Programa del servidor iniciado.*

Para iniciar el prototipo, primero hay que conectar el termostato a la red wifi que se desee. Para ello, se debe conectar el módulo wifi a un ordenador mediante un cable USB y ejecutar la aplicación de configuración del módulo wifi. Hay que tener en cuenta que el termostato tiene que estar apagado para que se envíen correctamente los comandos por el cable USB desde la aplicación de configuración. Una vez establecida la conexión ya se puede dar alimentación al termostato. En la siguiente imagen se muestra la aplicación de configuración del módulo wifi en ejecución.

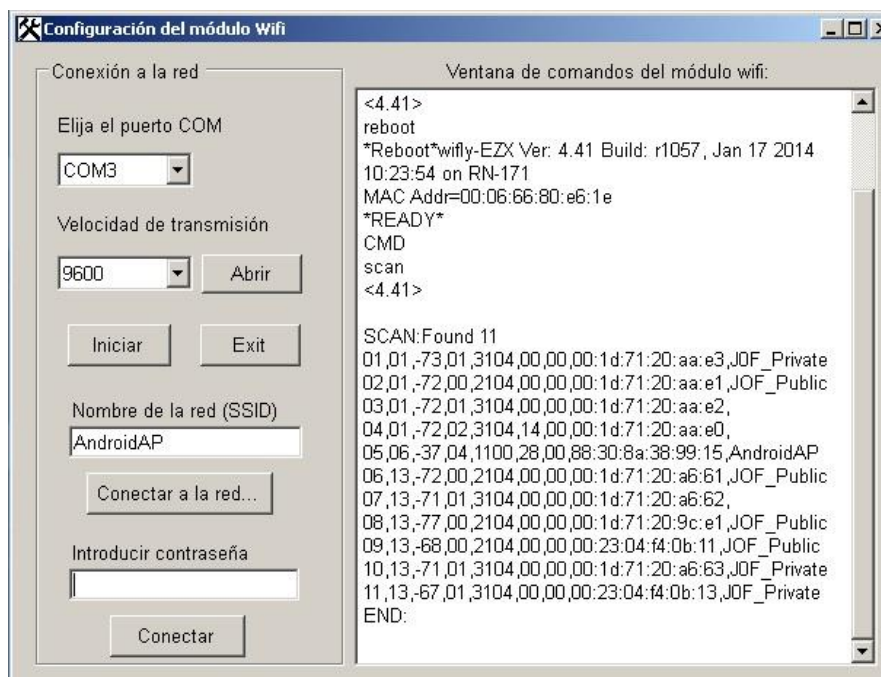


Figura 32. Aplicación de configuración del módulo wifi en ejecución.

En las siguientes imágenes se muestra el prototipo del termostato en funcionamiento. En la pantalla LCD se puede ver la temperatura que mide el sensor en ese momento y a la derecha se puede leer "OFF", que indica que la orden que está enviando el termostato al calentador es de apagado.



Figura 33. Prototipo del termostato en funcionamiento.



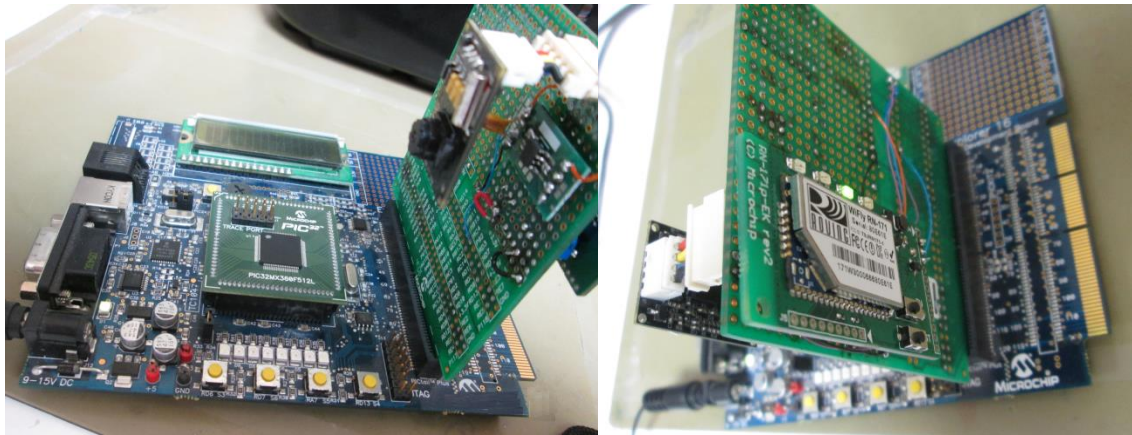


Figura 34. Imagen del termostato donde se ve la cámara de fotos y el módulo wifi.

En la siguiente imagen se puede ver al servidor recibiendo información del termostato cíclicamente.

```
serv..tion_df31a5193da6f6a0_53e4f7cab840b6a7
Esperando a una conexion...
Esperando a una conexion...
Tamaño de foto: 21344 bytes
El numero de ID recibido del MODULO es 'ID1'
El password recibido del MODULO es: pass1
La temperatura real recibida del MODULO es: 24
Enviados 6 bytes al cliente.
Se ha cerrado la conexion con el cliente.
Esperando a una conexion...
Tamaño de foto: 21040 bytes
El numero de ID recibido del MODULO es 'ID1'
El password recibido del MODULO es: pass1
La temperatura real recibida del MODULO es: 25
Enviados 6 bytes al cliente.
Se ha cerrado la conexion con el cliente.
```

Datos recibidos del termostato. (1º)

Datos recibidos del termostato. (2º)

Figura 35. Servidor recibiendo tramas del mismo termostato cíclicamente.

En las siguientes imágenes se puede ver la aplicación del usuario y la del servidor cuando la contraseña que envía el usuario no es la correspondiente al termostato al que se quiere asociar.

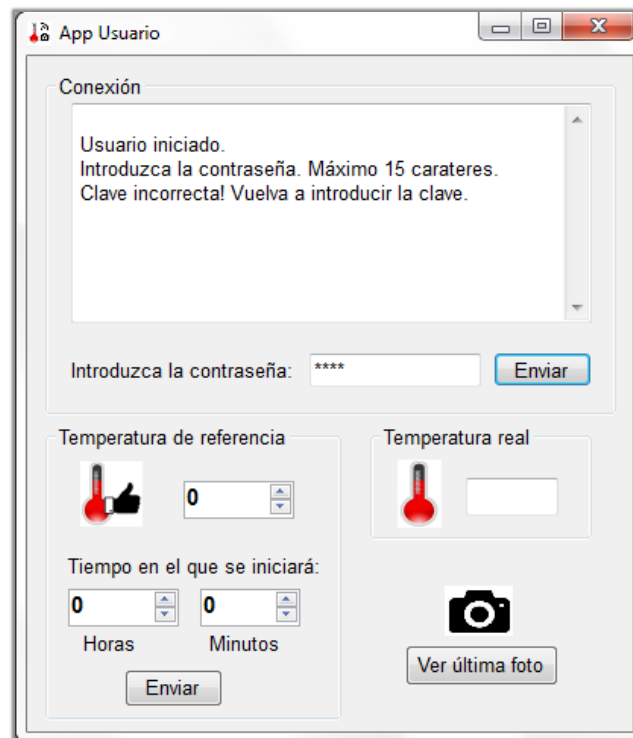


Figura 36. Aplicación del usuario cuando la contraseña enviada no es la correcta.

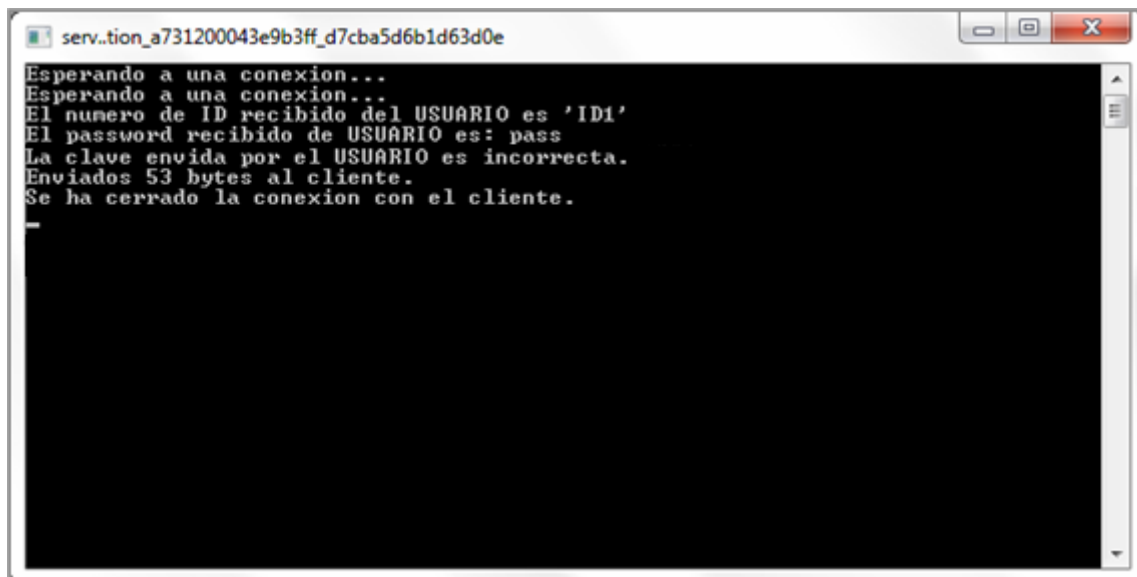


Figura 37. Servidor cuando un usuario envía una contraseña incorrecta desde su aplicación.

En la siguiente imagen se muestra la aplicación del usuario y el servidor cuando el usuario envía la contraseña correcta y el servidor envía los datos del termostato al que se ha asociado en la conexión.

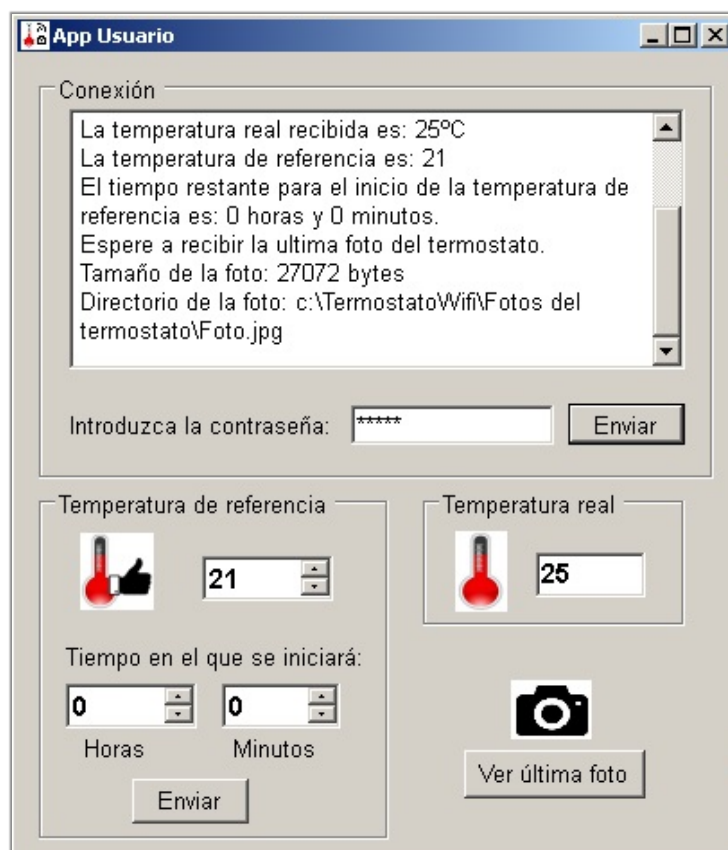


Figura 38. Aplicación del cliente al recibir los datos del termostato al que se ha asociado.

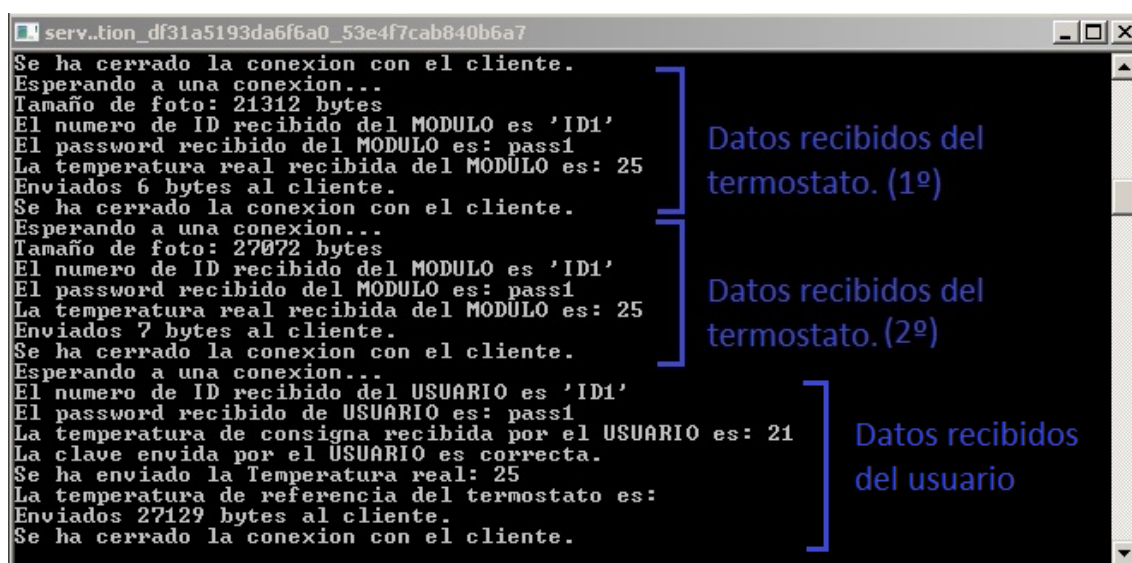


Figura 39. Servidor enviando los datos del termostato a la aplicación del usuario.

Una vez recibidos los datos en la aplicación del usuario, ya se podrá visualizar la última fotografía que el termostato haya enviado al servidor. En la siguiente imagen se puede ver cómo se ve un foto al pulsar el botón de ver última foto.

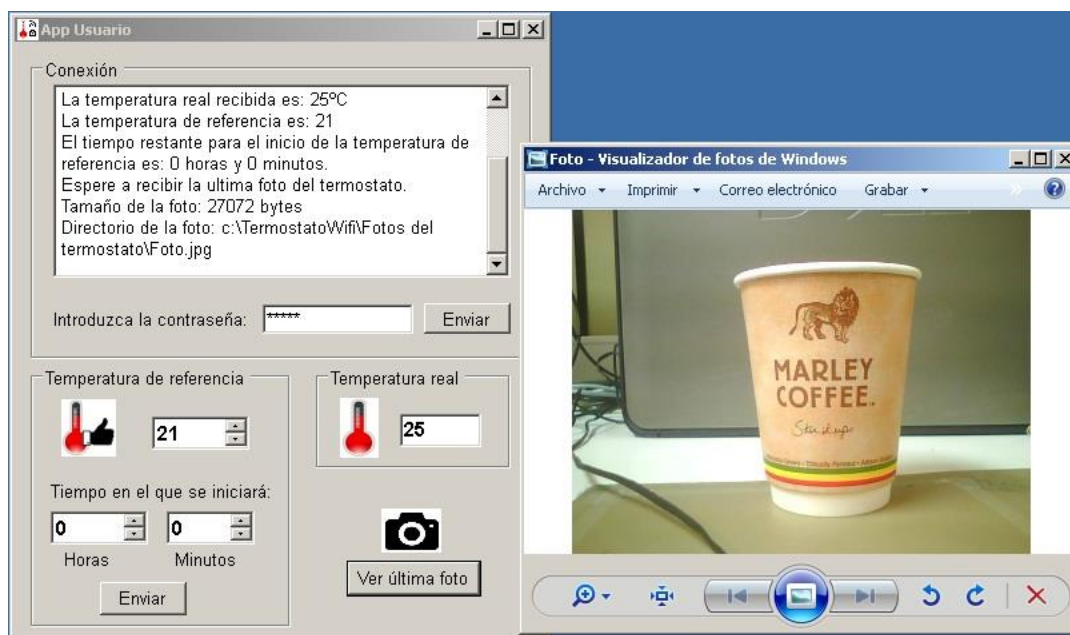


Figura 40. Visualización de una foto en la aplicación del usuario.

En la siguiente imagen se puede ver la aplicación del usuario cuando se envía una temperatura de referencia y un tiempo de retardo en el que se iniciará la nueva temperatura de referencia.

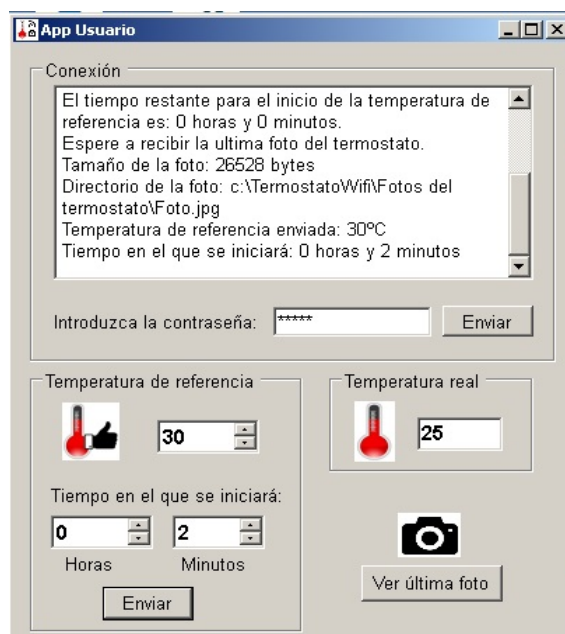


Figura 41. Aplicación de usuario cuando se envía una temperatura de referencia y un tiempo de retardo



La siguiente imagen muestra la capacidad de multigestión del servidor. Para comprobarlo, se ha programado el termostato wifi con el número de identificación "ID4" y contraseña "pass4" y se ha hecho que envíe una trama de datos al servidor. Luego, se ha vuelto a programar el termostato wifi, pero esta vez con el número de identificación "ID1" y contraseña "pass1" y se ha hecho que envíe una trama de datos al servidor. A continuación, se han ejecutado dos aplicación de usuario, una con el número de identificación "ID1" y la otro con "ID4". Al enviar la contraseña "pass1" con la aplicación de usuario ID1 se reciben los datos del termostato correspondiente y al enviar la contraseña "pass4" con la aplicación de usuario ID4 se reciben los datos del termostato correspondiente.

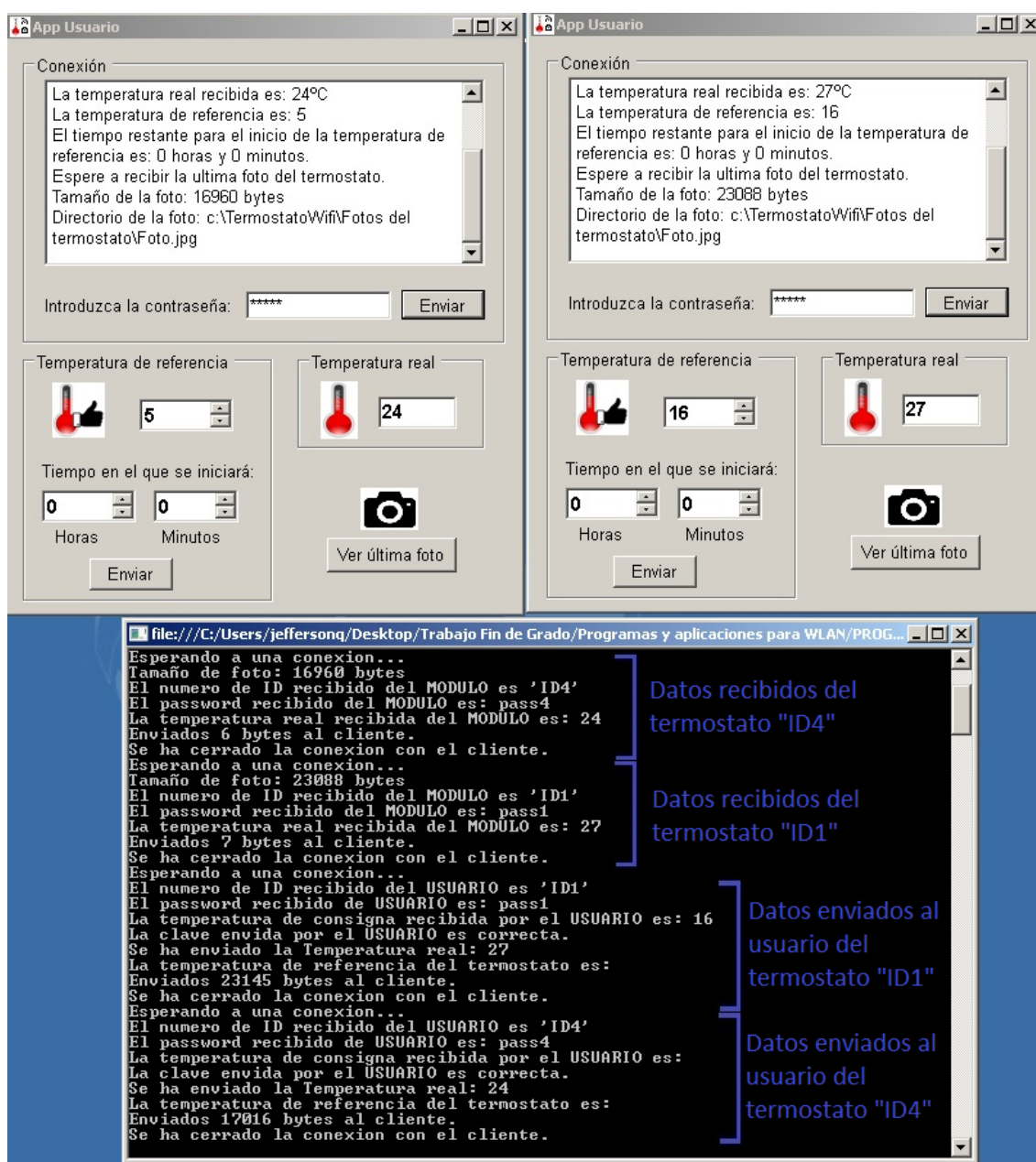


Figura 42. Muestra de la capacidad de multigestión del servidor.

## 11. BIBLIOGRAFÍA

### Páginas web:

- (1) [www.microchip.com](http://www.microchip.com)
- (2) <http://www.msdn.microsoft.com/>
- (3) <http://www.matrix.es/>
- (4) <https://es.wikipedia.org/>

### Libros:

- (5) Guía de referencia C#3.0. Autor: Jay Hilyard y Stephen Teilhet. Editorial: Anaya / O'REILLY
- (6) Programming 32-bit Microcontrollers in C. Autor: Lucio Di Jasio. Editorial: Newnes.
- (7) Programación en C. Autor: Byron Gottfried